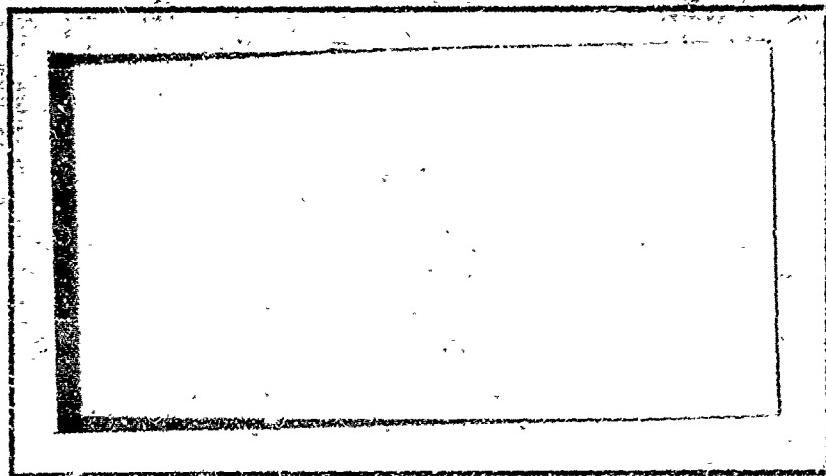
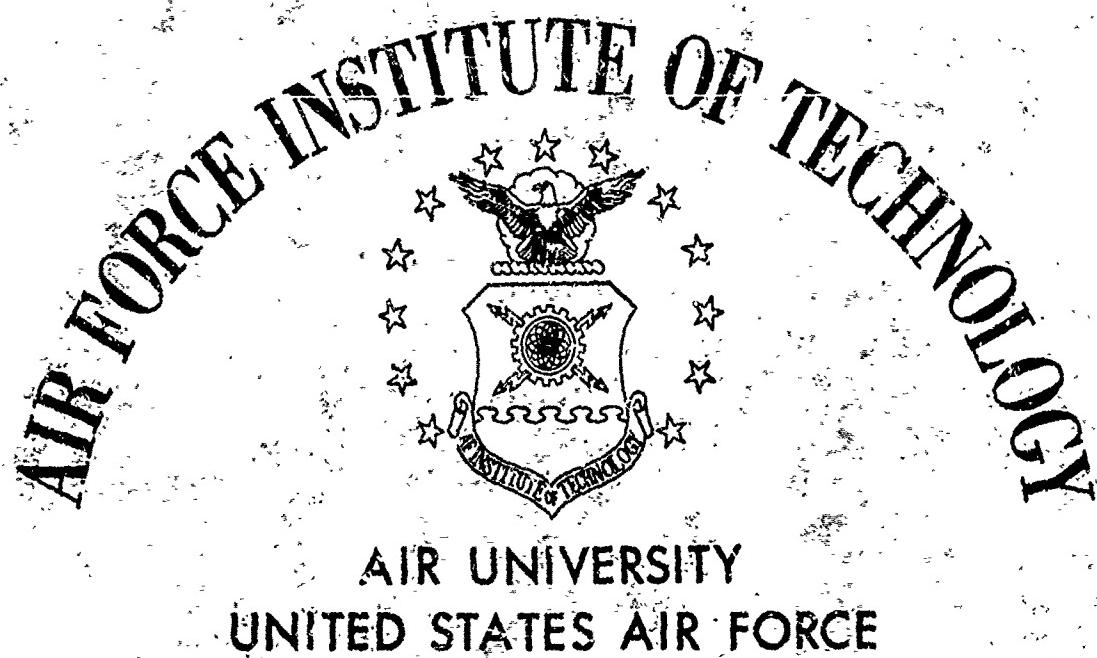


AU TWO.

ONE FILE COPY

ID A 053267



SCHOOL OF ENGINEERING

WRIGHT-PATTERSON AIR FORCE BASE, OHIO

D D C
REPRODUCED
APR 28 1978
REF ID: A6567V-50
A

AFIT/GA/EE/77-5

1

AD A 053267

AD No.

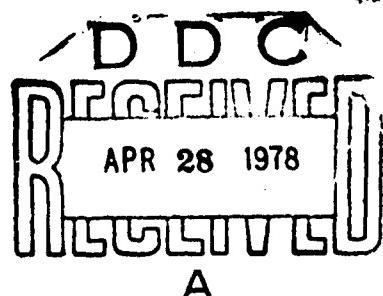
DDC FILE COPY

OPTIMAL MISSILE EVASION

THESIS

AFIT/GA/EE/77-5

Robert L. Smith
Capt USAF



Approved for public release; distribution unlimited

AFIT/GA/EE/77-5

OPTIMAL MISSILE EVASION

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by
Robert L. Smith, B.S.
Capt USAF
Graduate Astronautical Engineering
December 1977

A

AFIT/GA/EE/77-5	White Section <input checked="" type="checkbox"/>
R&G	Buff Section <input type="checkbox"/>
ASR	<input type="checkbox"/>
drawn 12-77	<input type="checkbox"/>
Approved for public release; distribution unlimited.	
APPROVAL CODES	
REF ID: A	
1 2 3 4 5 6 7 8 9 10	

Approved for public release; distribution unlimited.

Preface

This thesis represents the result of a ten month investigation of a second order technique for finding the optimum controls to be used by an aircraft evading an air-to-air heat seeking missile. The approach involves the application of optimal control theory utilizing a Differential Dynamic Programming Model to determine the optimum controls.

This thesis was sponsored by the Aeronautical Systems Division as part of a study being conducted by the Air Force for a better visual display to the pilot for evading an air-to-air missile.

I have had great satisfaction in developing the algorithm for this project. My only disappointment is that I was unable to finish the algorithm in order to obtain the desired results; however, I do feel that I have laid the groundwork that could lead to a worthwhile second order technique.

I wish to sincerely thank my thesis advisor, Major James Funk, for his assistance and guidance during this project. I would also like to thank my sponsor, Mr. Mike Breza, for his helpful suggestions.

I dedicate this thesis to my wife, Sharon, and my children, Todd and Kristi, who gave me encouragement and understanding, and exercised an incredible amount of patience.

Robert Smith

Contents

	<u>Page</u>
Preface	ii
List of Figures	v
Glossary	vi
Abstract	vii
I. Introduction	1
Purpose of the Study	1
Background	1
Scope	2
Assumptions	3
General Approach	4
II. The State Equations	6
General Description	6
Defining the State Vector	6
Derivation of Force Equations	7
Required Aerodynamic Acceleration of the Missile	11
Angle of Attack of the Missile	15
Bank Angle of the Missile	15
Defining the Nonlinear Differential State Equations	17
III. Terminal Cost	19
IV. Terminal Constraint	26
V. Differential Dynamic Programming Equations .	30
VI. Computational Procedure	35
Step 1 - Backward Integration of the State Equations Using Nominal Control . .	36
Step 2 - Forward Integration of the State Equations	40
Step 3 - Backward Integration with Unconstrained Terminal Condition	40
Step 4 - Forward Integration of State Equations Using Improved Control	45
Step 5 - Backward Integration with Constrained Terminal Condition	48
Step 6 - Forward Integration of State Equations Using Optimum Control	50

Contents

	<u>Page</u>
VII. Results	52
VIII. Conclusions and Recommendations	58
Bibliography	60
Appendix A: Detailed Schematic of the Program . . .	61

List of Figures

<u>Figure</u>		<u>Page</u>
1	Velocity Vector in Inertial Frame	8
2	Velocity Axis Coordinate System With Angle of Attack	10
3	Velocity Axis Coordinate System With Bank Angle	10
4	Determination of Gravity Components in Velocity Reference Frame	13
5	Determination of Missile Bank Angle	16
6	Terminal Cost Geometry	20
7	Terminal Constraint Geometry	27
8	Differential Dynamic Programming Model Flow Chart	37
9	Initialization of Terminal Constraint	39
10	Detailed Flow Chart of the Program	62

Glossary

α_m	Angle of attack of missile
β_m	Bank angle of missile
CD_{om}	Parasite drag coefficient of missile
CD_{oT}	Parasite drag coefficient of target
$CL_{\alpha m}$	Lift curve slope of missile
$CL_{\alpha T}$	Lift curve slope of target
K_m	Induced drag coefficient for missile
K_T	Induced drag coefficient for target
m_m	Mass of missile
m_T	Mass of target
n	Proportionality constant of missile
S_m	Surface area of missile
S_T	Surface area of target
t_f	Fuzing time of missile
T_f	Fuzing delay interval for missile
T_m	Thrust of missile
τ_p	Time constant of pitch attitude response of target
τ_T	Time constant of thrust response of target

Abstract

The purpose of the study is to formulate a method to determine the control strategies that maximize the probability of survival for an evading aircraft. This is equivalent to minimizing the probability of kill for the attacking air-to-air missile. The controls of the evading aircraft consist of the commanded angle of attack, bank angle, and the commanded coefficient of thrust. The missile model developed is a typical air-to-air infrared missile using proportional navigation steering. The probability of kill is modeled as ellipsoidal iso-cost surfaces with a cost value that decays exponentially as the ellipsoid size increases. The flattening and position of the ellipsoid centroid account for the shape, orientation and vulnerability of the aircraft. The problem terminates when the line-of-sight from the missile to the target aligns with the surface of the missile's fusing cone. The algorithm developed employs a second order differential dynamic programming model for optimizing the controls of the evading aircraft.

OPTIMAL MISSILE EVASION

I. Introduction

Purpose of the Study

With the newer fighter aircraft and fire control system a problem exists for both attacking and evading aircraft. A method of determining and displaying real time probability of kill (P_k) to the pilot of the attacking aircraft using an air-to-air missile is very desirable. On the other hand, the effectiveness of missiles against aircraft has made it extremely desirable to provide the pursued pilot with aids for evading missiles. This study will be concerned with the latter of these two problems, that is the problem associated with the evading aircraft.

The evasion problem is a two step process consisting of:

1. Determining the optimum control strategies to be used by the evading pilot.
2. Providing on-board computation and display of cues or solutions for the pilot.

This study represents the first step in that process, that is determining the optimum controls to be used by the evading pilot.

Background

The maximum and minimum effective ranges of air-to-air

missiles are functions of attacker state, target state, and performance characteristics of both vehicles. Operational experience has demonstrated that pilots have difficulty in accurately estimating valid launch conditions during an encounter. Consequently, effective employment of air-to-air missiles requires two distinct functions: 1) an accurate missile launch envelope computation performed in an airborne computer, and 2) display of appropriate parameters to the attacking pilot so that he can recognize and take advantage of valid launch opportunities.

The missile-launch envelope considerations are really beyond the scope of this study, which concentrates on evasion. However, the solutions to the evasion problem could be used as a basis to determine launch envelopes. It is still a sizeable step to solve and process the necessary solution data in order to obtain usable probability of kill information.

Present airborne digital computers, such as in the F-15, have made it practical to develop and to implement evading strategy computations and to display these parameters to the pilot. Thus this study is concerned with the development of improved strategies to be used by an evading target; solutions that consider more information than past methods.

Scope

The state variable equations are simplified where ever possible without sacrificing any significant realism. There is not as much freedom of motion as in the actual case due to

the assumption of coordinated turns by the aircraft.

The set of controls were chosen such that the evading aircraft has flexibility in controls and is still realistic.

The terminal cost function corresponds generally to a detailed simulation of the end game which incorporates the shape, orientation and vulnerability of the aircraft. Individual component vulnerabilities of a particular aircraft are not considered, only the general characteristics of a typical aircraft.

The second order differential dynamic programming algorithm was selected for evaluation in solving this problem. This choice was based on the high degree of nonlinearity of the state equations, and convergence difficulties with a first order algorithm. The algorithm was adapted to this problem using unspecified final time.

This report will be limited to an approach for obtaining the following information:

1. The trajectory of the missile.
2. The optimum trajectory of the evading target.
3. The minimum P_k of the missile.
4. The optimum controls used by the evading aircraft.

The information obtained is determined from a specific set of launch conditions.

Assumptions

Certain assumptions can be made to reduce the complexity of the problem without significantly affecting the character

of the solution. They are as follows:

1. Rigid body dynamic models for both the missile and the evading target.
2. The yaw angle is assumed negligible.
3. For the evader, the angle of attack and thrust responses to respective commands are modeled as linear first order systems with appropriate time constants.
4. The bank-angle time responses are assumed rapid enough to neglect any time delay in the response.
5. The missile fusing cone angle is fixed, as is the fusing delay time.
6. The density of the atmosphere as a function of altitude is given by:

$$\rho = \rho_0 e^{-Z/Z_0}$$

where

$$\rho_0 = .0023769 \text{ slugs/ft}^3$$

$$Z_0 = 23800 \text{ ft}$$

Z = the altitude above the earth's surface.

General Approach

The material is presented in the following order. First the equations of motion for both the missile and evading aircraft are derived. The proportional navigation steering is

then developed along with the computed acceleration for the missile. The state equations for the dynamic model are next outlined. This is followed by the derivation of the cost function derived from the probability of kill geometry. Next the terminal constraint geometry is obtained. The required equations for the differential dynamic programming algorithm (Ref 1:47) are outlined. Finally, the computational procedure used for the algorithm is discussed, followed by results and conclusions.

II. The State Equations

General Description

The continuous-time dynamic system modeled for this program is described by the following set of nonlinear ordinary differential equations:

$$\begin{aligned}\dot{x}_T &= f(x_T, u) \quad ; \quad x_T(t_0) = x_{T_0} \\ \dot{x}_m &= f(x_m) \quad ; \quad x_m(t_0) = x_{m_0}\end{aligned}\tag{1}$$

where the subscripts T and m denote target states and missile states respectively. The performance of the system is measured by minimization of a terminal cost function given as:

$$P_k = e^{-X^T(t_f) Q X(t_f)}\tag{2}$$

subject to the terminal constraint of the form:

$$\psi(x_m(t_f), x_T(t_f)) = 0\tag{3}$$

where the final time t_f is given implicitly.

Defining the State Vector

The state vector for this optimal missile evasion problem includes the distance components between the target and missile, the velocity of both target and missile, the heading of target and missile, the flight path angle of target and missile, the angle of attack of the target, and the coefficient of thrust of the target. In standard notation the

state vector is defined as:

$$\underline{x} = \begin{bmatrix} x_m - x_T \\ y_m - y_T \\ z_m \\ z_T \\ v_m \\ v_T \\ \psi_m \\ \psi_T \\ \gamma_m \\ \gamma_T \\ \alpha_T \\ c_T \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} \quad (4)$$

The controls determined for this problem are the commanded angle of attack of the target, the bank angle of the target, and the commanded coefficient of thrust of the target. In vector form the controls are designated as:

$$\underline{u} = \begin{bmatrix} \alpha_{Tc} \\ \beta_T \\ c_{Tc} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (5)$$

where the subscript c denotes commands.

Derivation of Force Equations

Both missile and evading aircraft are modeled as rigid

bodies with their velocity vectors expressed in an inertial reference frame (*ijk*) as depicted in Fig. 1.

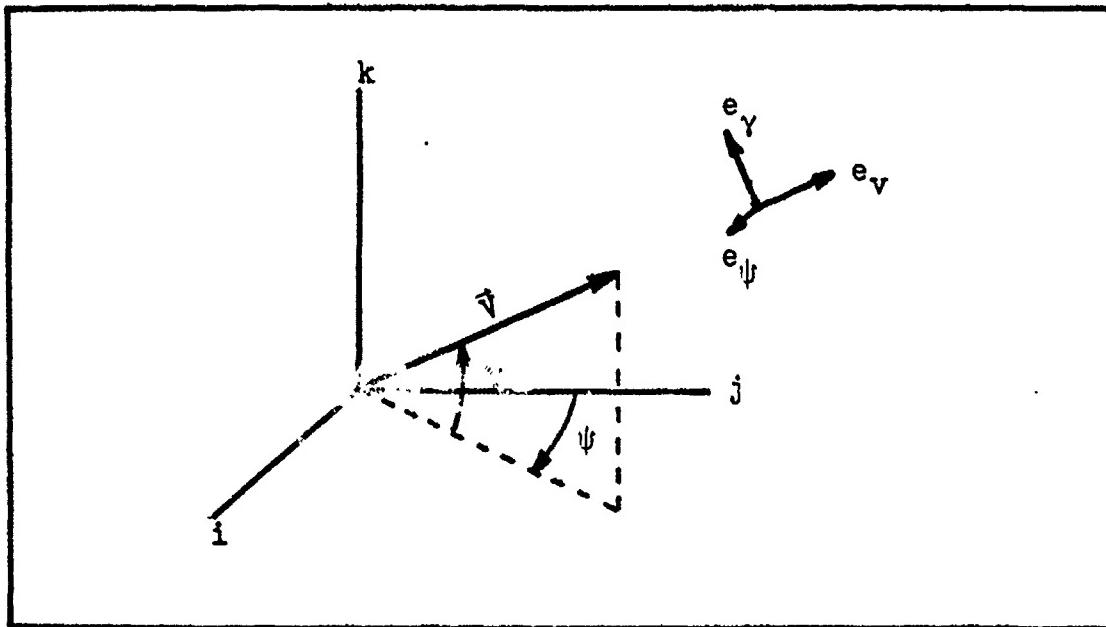


Fig. 1. Velocity Vector in Inertial Frame

The velocity axis frame is designated as \vec{e}_v , \vec{e}_ψ , \vec{e}_γ . The velocity of the center of mass in the velocity axis system is

$$\vec{V} = V\vec{e}_v \quad (6)$$

transformation into the inertial reference frame yields

$$\begin{aligned} \dot{X} &= V\cos(\gamma) \sin(\psi) \\ \dot{Y} &= V\cos(\gamma) \cos(\psi) \\ \dot{Z} &= V\sin(\gamma) \end{aligned} \quad (7)$$

where *X* and *Y* are horizontal position coordinates, *Z* is altitude, *V* is speed, γ is the flight path angle and ψ is the

heading angle with respect to the Y axis.

In order to obtain the accelerations of each vehicle, the derivative of the velocity in the moving frame must be taken.

$$\dot{\vec{v}} = \dot{v}\vec{e}_v + v\dot{\vec{e}}_v \quad (8)$$

and recalling that

$$\dot{\vec{e}}_v = \vec{\omega} \times \vec{e}_v \quad (9)$$

where the angular rate $\vec{\omega}$ is given as

$$\vec{\omega} = \dot{\gamma}\vec{e}_\psi - \dot{\psi}\vec{e}_z \quad (10)$$

and where

$$\vec{e}_z = \cos(\gamma)\vec{e}_y + \sin(\gamma)\vec{e}_v \quad (11)$$

therefore, combining Eqs (9), (10), and (11) and substituting into Eq (8) yields

$$\dot{\vec{v}} = \dot{v}\vec{e}_v + v\dot{\gamma}\vec{e}_y + v\dot{\psi}\cos(\gamma)\vec{e}_\psi \quad (12)$$

The acceleration of the body times the mass of the body is equal to the sum of the external forces, or

$$m\dot{\vec{v}} = \vec{F} \quad (13)$$

where \vec{F} is composed of thrust, lift, drag, and gravity. A more detailed representation of the velocity axis coordinate system with the forces acting on the vehicle is shown in

Fig. 2 and Fig. 3.

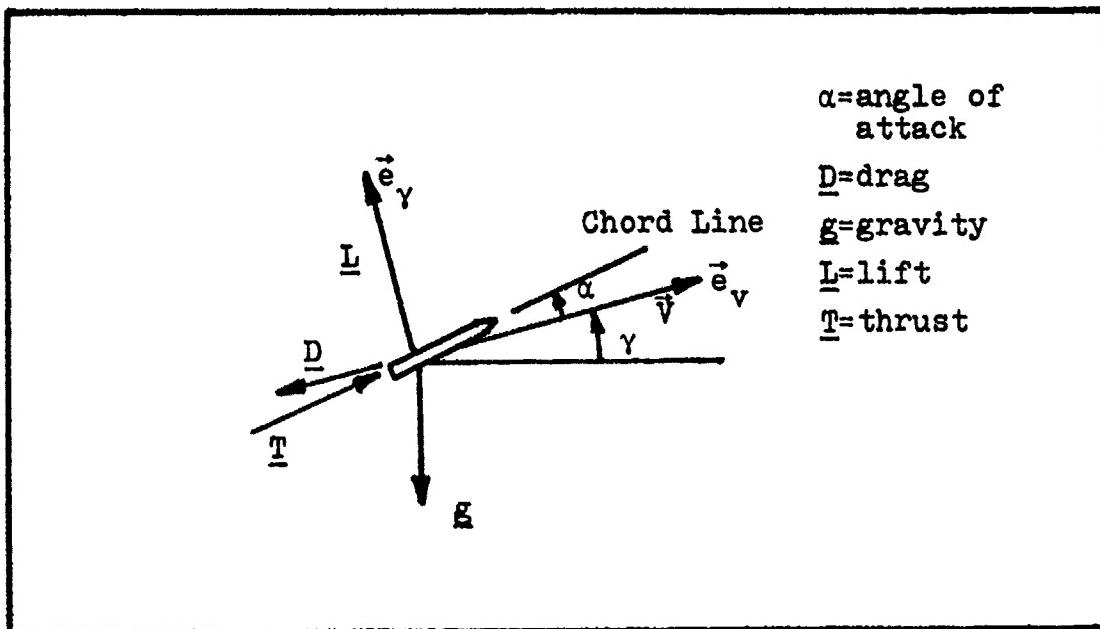


Fig. 2. Velocity Axis Coordinate System With Angle of Attack

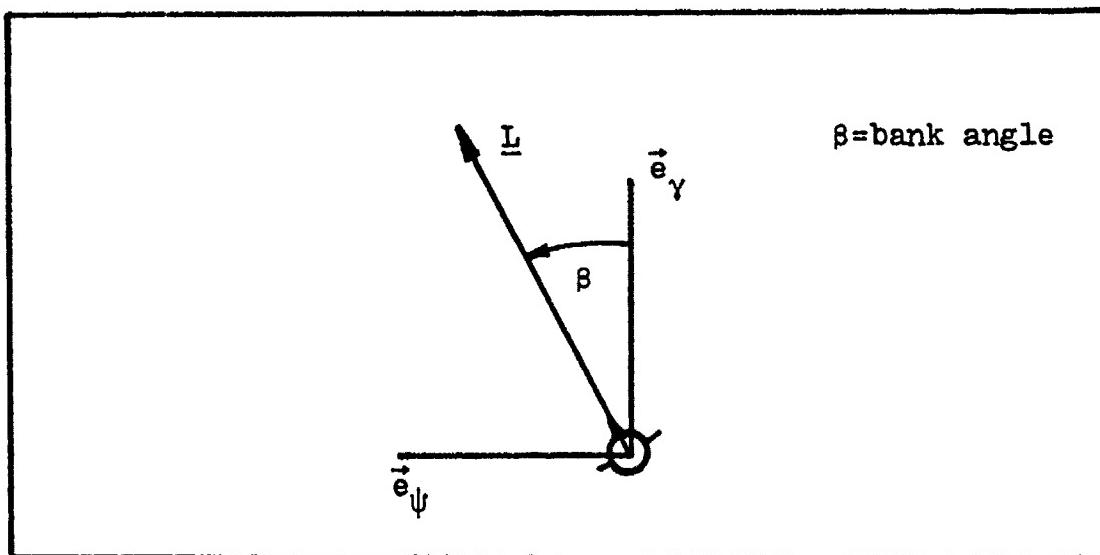


Fig. 3. Velocity Axis Coordinate System With Bank Angle

Referring to Fig. 2 and Fig. 3 the following force components are obtained:

$$\begin{aligned}
 \text{Drag} &= -D \vec{e}_v \\
 \text{Thrust} &= T \cos(\alpha) \vec{e}_v + T \sin(\alpha) \vec{e}_\gamma \\
 \text{Lift} &= L \cos(\beta) \vec{e}_\gamma + L \sin(\beta) \vec{e}_\psi \\
 \text{Gravity} &= -g \cos(\gamma) \vec{e}_\gamma - g \sin(\gamma) \vec{e}_v
 \end{aligned} \tag{14}$$

where

$$\begin{aligned}
 D &= 1/2 \rho v^2 S [C_d_0 + K(C_L \alpha)^2] \\
 L &= 1/2 \rho v^2 S C_L \alpha
 \end{aligned} \tag{15}$$

Combining Eqs (12) through (15) the following acceleration terms are obtained in the inertial axis coordinate system:

$$\begin{aligned}
 \dot{v} &= (T \cos(\alpha) - 1/2 \rho S v^2 (C_d_0 + K(C_L \alpha)^2)) / m \\
 &\quad - g \sin(\gamma) \\
 \dot{\gamma} &= 1/2 \rho S V (C_L \alpha) \cos(\beta) / m + \frac{T \sin(\alpha)}{m V} \\
 &\quad - g \cos(\gamma) / V \\
 \dot{\psi} &= 1/2 \rho S V (C_L \alpha) \sin(\beta) / m \cos(\gamma)
 \end{aligned} \tag{16}$$

Required Aerodynamic Acceleration of the Missile

Proportional navigation provides a rate of change of the missile heading directly proportional to the rate of rotation of the line-of-sight from the missile to the target (providing the rates are within missile performance limitations). The line-of-sight rate from the missile to the

target is defined as

$$\dot{\ell}_r = \frac{\underline{r} \times \underline{v}_r}{r^2} \quad (17)$$

where $r^2 = \underline{r} \cdot \underline{r}$ and \underline{r} is the relative distance between the missile and target, therefore:

$$\underline{r} = (x_T - x_m)\hat{i} + (y_T - y_m)\hat{j} + (z_T - z_m)\hat{k} \quad (18)$$

and \underline{v}_r is the relative velocity between the missile and target, therefore:

$$\begin{aligned} \underline{v}_r = & (v_T \cos(\gamma_T) \sin(\psi_T) - v_m \cos(\gamma_m) \sin(\psi_m))\hat{i} \\ & + (v_T \cos(\gamma_T) \cos(\psi_T) - v_m \cos(\gamma_m) \cos(\psi_m))\hat{j} \\ & + (v_T \sin(\gamma_T) - v_m \sin(\gamma_m))\hat{k} \end{aligned} \quad (19)$$

The desired turn rate for the missile is then equal to a proportional navigation constant (n) times the line-of-sight rate, or the desired turn rate is $n \dot{\ell}_r$. For a typical air-to-air missile n is in the range of 2 to 4; and for this problem n is set equal to 3. The computed acceleration of the missile, defined as \underline{a}_c , is then given as:

$$\underline{a}_c = n \dot{\ell}_r \times \underline{v}_m \quad (20)$$

In order to determine the required aerodynamic acceleration of the missile, designated as \underline{a}_n , the effect of gravity must be incorporated. Therefore, combining Eq (20) with the effect of gravity the required aerodynamic acceleration becomes

$$\underline{a}_n = \underline{a}_c - \underline{g}_n \quad (21)$$

where \underline{a}_n is the required aerodynamic acceleration and \underline{g}_n is the component of gravity normal to the velocity of the missile. \underline{a}_n is the aerodynamic acceleration which the vehicle should produce in order to obtain the proper normal acceleration in the presence of gravity. Referring to Fig. 4, gravity in the velocity frame system is given as:

$$\underline{g}_{\text{velocity}} = -g \sin(\gamma) \vec{e}_v - g \cos(\gamma) \vec{e}_y + 0 \vec{e}_\psi \quad (22)$$

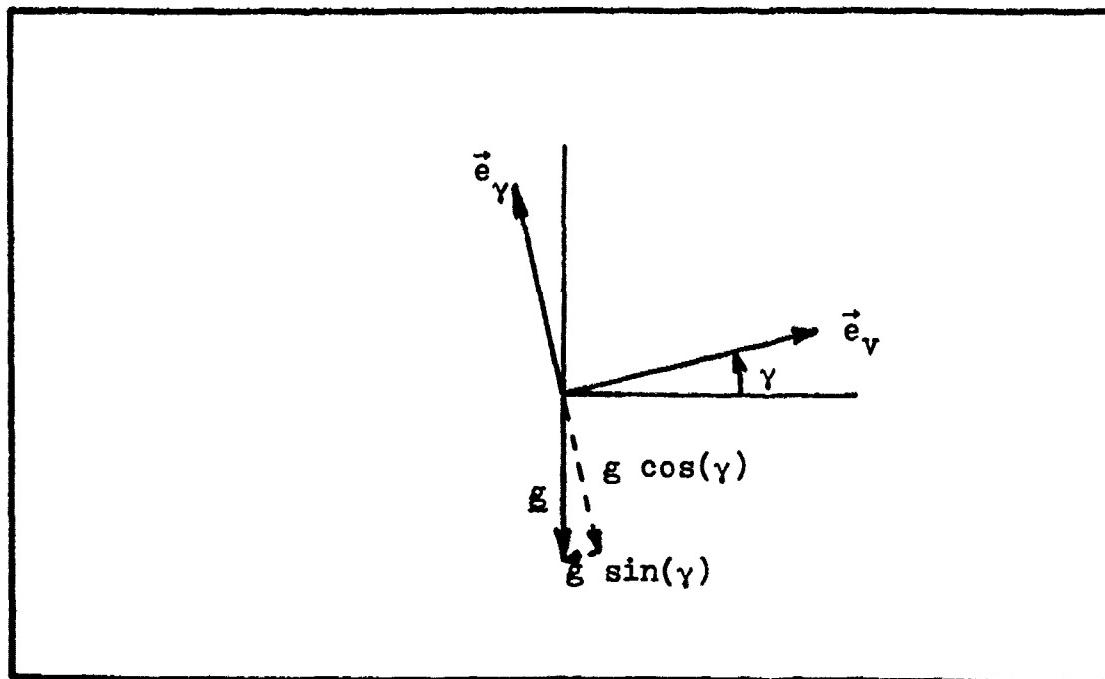


Fig. 4. Determination of Gravity Components in Velocity Reference Frame

In the required aerodynamic acceleration formula, the effective component of gravity is that component which is normal to the velocity vector; or from Eq (22) effective component of gravity becomes:

$$\begin{aligned} \text{g velocity} &= -g \cos(\gamma) \vec{e}_\gamma \\ &\text{frame} \\ &\text{effective} \end{aligned} \quad (23)$$

Referring to Fig. 1, transformation from the velocity coordinate system to the inertial coordinate system is given by:

$$\begin{bmatrix} e_\psi \\ e_v \\ e_\gamma \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & \sin(\gamma) \\ 0 & -\sin(\gamma) & \cos(\gamma) \end{bmatrix} \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ k \end{bmatrix}$$

which simplifies to:

$$\begin{bmatrix} e_\psi \\ e_v \\ e_\gamma \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \cos(\gamma) \sin(\psi) & \cos(\gamma) \cos(\psi) \sin(\gamma) & \cos(\gamma) \cos(\psi) \cos(\gamma) \\ -\sin(\gamma) \sin(\psi) & -\sin(\gamma) \cos(\psi) \sin(\gamma) & -\sin(\gamma) \cos(\psi) \cos(\gamma) \end{bmatrix} \begin{bmatrix} i \\ j \\ k \end{bmatrix} \quad (24)$$

Combining Eqs (23) and (24) the effective component of gravity in the inertial coordinate system then becomes:

$$\begin{aligned} \text{g inertial} &= \text{g}_n = g \cos(\gamma) \sin(\gamma) \sin(\psi) \vec{i} \\ &\text{frame} \\ &\text{effective} & + g \cos(\gamma) \sin(\gamma) \cos(\psi) \vec{j} \\ & & - g \cos^2(\gamma) \vec{k} \end{aligned} \quad (25)$$

Therefore, Eq (25) along with the computed acceleration of

the missile (Eq 20) determines the required aerodynamic acceleration of the missile.

Angle of Attack of the Missile

A detailed representation of the forces acting on the missile is depicted in Fig. 2. Equating forces and noting that the thrust vector T acts along the vehicle axis of symmetry, the force normal to the flight path is given as:

$$F_{\text{normal}} = m a_n = \text{Lift} + T \sin(\alpha) \quad (26)$$

Assuming that the angle of attack remains small so that $\sin(\alpha)$ approximately equals α , and substituting in for lift, Eq (15), the force normal to the flight path becomes:

$$m a_n = 1/2 \rho v^2 S C_{L\alpha} \alpha + T \alpha \quad (27)$$

Solving Eq (27) for the angle of attack of the missile yields

$$\alpha_m = \frac{a_n m}{1/2 \rho v^2 S C_{L\alpha} \alpha + T} \quad (28)$$

Bank Angle of the Missile

In order to determine the bank angle of the missile, refer to Fig. 5 for a schematic of the bank angle in the velocity coordinate system. The missile bank angle, β_m , is the angle between the unit vector \vec{e}_y and the vector \vec{u}_a where \vec{u}_a is the unit vector along the direction of the required aerodynamic acceleration a_n . Using the law of cosines, the

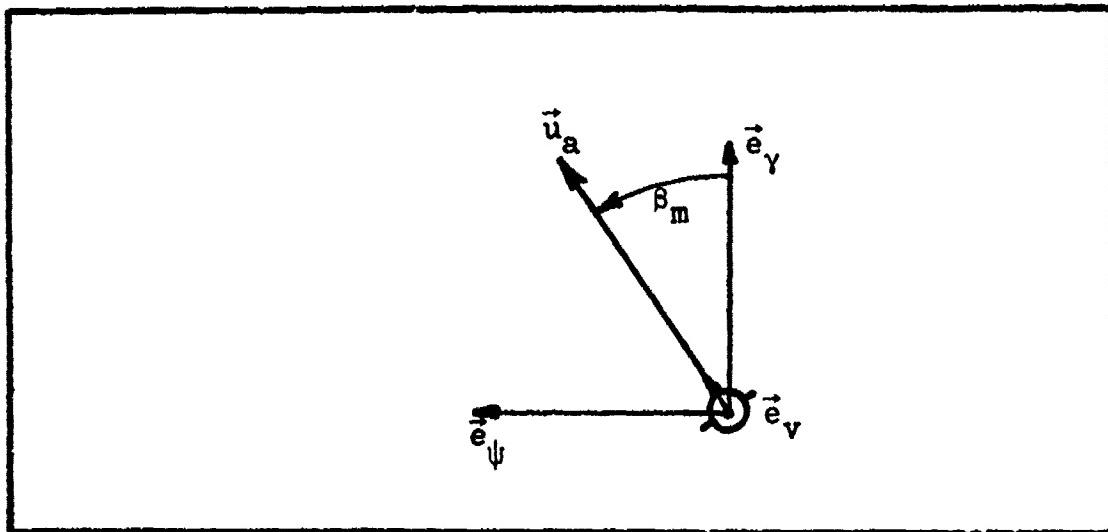


Fig. 5. Determination of Missile Bank Angle

bank angle is then determined by:

$$\beta_m = \pm \cos^{-1}(\vec{e}_\gamma \cdot \vec{u}_a) \quad (29)$$

where the unit vector \vec{u}_a is determined by:

$$\vec{u}_a = \frac{\vec{a}_n}{|\vec{a}_n|} \quad (30)$$

The third component of Eq (24) gives \vec{e}_γ

$$\vec{e}_\gamma = -\sin(\gamma) \sin(\psi) \vec{i} - \sin(\gamma) \cos(\psi) \vec{j} + \cos(\gamma) \vec{k} \quad (31)$$

Referring to Fig. 5 and Eq (29), the following stipulations are made on the bank angle:

1. If $\vec{e}_\psi \cdot \vec{u}_a > 0$ then β_m is between 0 and 180 degrees,

2. If $\vec{e}_\psi \cdot \vec{u}_a < 0$ then β_m is between 180 and 360 degrees,

where the unit vector \vec{e}_ψ is given as:

$$\vec{e}_\psi = \cos(\psi)\vec{i} - \sin(\psi)\vec{j} \quad (32)$$

In other words, the positive sign of β_m is used when condition 1. is satisfied, and the negative sign of β_m is used when condition 2. is satisfied.

Defining the Nonlinear Differential State Equations

The velocity equations represented by Eq (7) and the acceleration equations represented by Eq (16) produce the following set of nonlinear ordinary differential state equations used in this program:

$$\dot{x}_m - \dot{x}_T = v_m \cos(\gamma_m) \sin(\psi_m) - v_T \cos(\gamma_T) \sin(\psi_T)$$

$$\dot{y}_m - \dot{y}_T = v_m \cos(\gamma_m) \cos(\psi_m) - v_T \cos(\gamma_T) \cos(\psi_T)$$

$$\dot{z}_m = v_m \sin(\gamma_m)$$

$$\dot{z}_T = v_T \sin(\gamma_T)$$

$$\begin{aligned} \dot{v}_m &= (T_m \cos(\alpha_m) - 1/2 \rho_m v_m^2 S_m (C_d_{0m} \\ &\quad + K_m (CL_{am} \alpha_m)^2)) / m_m - g \sin(\gamma_m) \end{aligned}$$

$$\begin{aligned} \dot{v}_T &= 1/2 \rho_T v_T^2 S_T (C_T \cos(\alpha_T) - (C_d_{0T} \\ &\quad + K_T (CL_{at} \alpha_T)^2)) / m_T - g \sin(\gamma_T) \end{aligned}$$

$$\dot{\psi}_m = 1/2 \rho_m S_m v_m CL_{am} \alpha_m \sin(\gamma_m) / m_m \cos(\gamma_m)$$

$$\begin{aligned}
\dot{\psi}_T &= 1/2 \rho_T S_T V_T CL_{\alpha T} \alpha_T \sin(s_T)/m_T \cos(\gamma_T) \\
\dot{\gamma}_m &= 1/2 \rho_m S_m V_m CL_{\alpha m} \alpha_m \cos(s_m)/m_m \\
&\quad + T_m \sin(\alpha_m)/m_m V_m - g \cos(\gamma_m)/V_m \\
\dot{\gamma}_T &= 1/2 \rho_T S_T V_T (CL_{\alpha T} \cos(s_T) \alpha_T/m_T \\
&\quad + C_T \sin(\alpha_T)/m_T) - g \cos(\gamma_T)/V_T \\
\dot{\alpha}_T &= (\alpha_{T_c} - \alpha_T)/\tau_p \\
\dot{C}_T &= (C_{T_c} - C_T)/\tau_T
\end{aligned} \tag{33}$$

The subscripts T and m denote the target and missile parameters respectively, τ_p is the time constant for the angle of attack response, and τ_T is the time constant for the thrust response of the target. α_m and s_m have previously been defined by Eqs (28) and (29), α_{T_c} , s_T , and C_{T_c} are the control variables defined by Eq (5), and where ρ_m and ρ_T are defined by:

$$\begin{aligned}
\rho_m &= \rho_0 e^{(-z_m/z_0)} \\
\rho_T &= \rho_0 e^{(-z_T/z_0)}
\end{aligned} \tag{34}$$

III. Terminal Cost

The objective of the terminal cost function is to make the model correspond to detailed simulation of the end game which incorporates the shape, orientation and vulnerability of the aircraft. Individual vulnerability of a particular aircraft is not considered, but only the generic character. The P_k ellipsoid parameters were chosen to fit detailed simulation results reasonably well. The general form of the terminal cost is given by:

$$P_k = e^{(-\underline{X}^T(t_f) Q \underline{X}(t_f))} \quad (2)$$

As previously stated, this optimization problem requires the minimization of this terminal cost. The terminal cost is a convex function described by ellipsoidal constant cost surfaces centered about a reference point near or on the aircraft. Each surface represents a constant value for the probability of kill for a particular missile. The probability of kill (P_k) values decrease exponentially with increasing concentric ellipsoid size. The ellipsoids can be considered fixed with respect to the target vehicle - moving and rotating with it. The P_k for each missile approach path is a function of the crossing aspect angles and the separation distance between the missile and target paths, as well as the relative velocity and altitude.

The P_k for each flight is evaluated by determining the

separation distance of the missile flight line projection from the target at the terminal time. The separation distance of the missile flight line projection is designated r in Fig. 6, and will be more precisely defined in the following paragraphs.

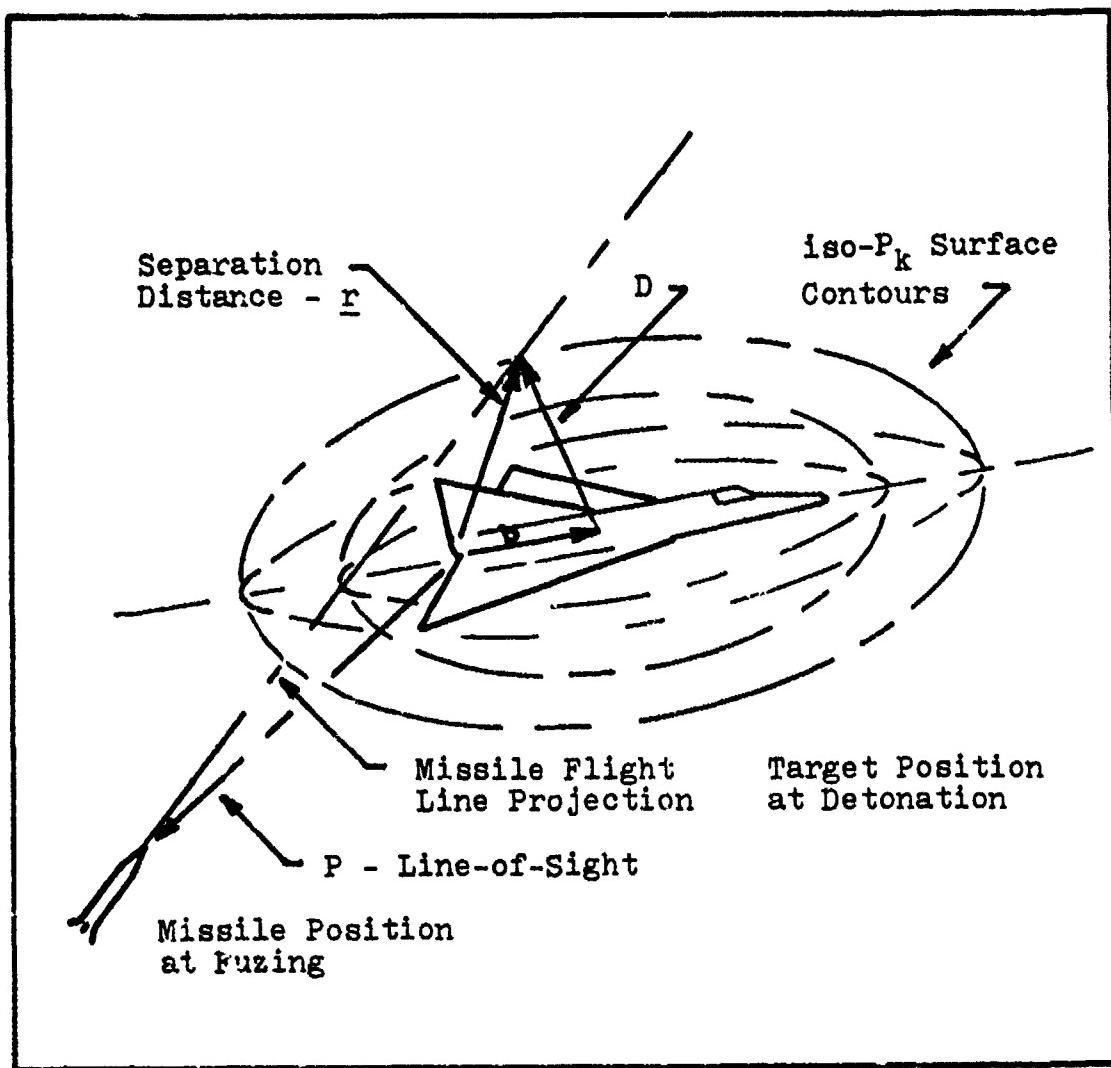


Fig. 6. Terminal Cost Geometry

Designating the fuzing time of the missile as t_f , which is the time when the target intercepts the missile fuzing cone and is the beginning of the missile delay interval, T_f , defined as the period of time from fuzing to detonation. The target position at detonation, designated as the origin $\underline{0}$, can be determined by

$$\underline{0} = \underline{x}_T(t_f) + \underline{v}_T(t_f) \cdot T_f \quad (35)$$

where \underline{x}_T is the target position at t_f and \underline{v}_T is the target velocity at t_f , in the inertial coordinate system. The separation distance \underline{r} is the offset distance of the missile flight path projection from the origin. The vector \underline{P} , which is the line-of-sight vector from the missile at fuzing to the target at detonation is then given by:

$$\underline{P} = \underline{x}_m(t_f) - \underline{0} \quad (36)$$

where $\underline{x}_m(t_f)$ is the position of the missile at the fuzing time. Substituting Eq (35) into Eq (36) yields

$$\underline{P} = \underline{x}_m(t_f) - \underline{x}_T(t_f) - \underline{v}_T(t_f) \cdot T_f \quad (37)$$

The vector \underline{r} , which is the vector from the origin perpendicular to the missile flight line projection, is a function of \underline{P} ,

$$\underline{r} = (\underline{u} \times \underline{P}) \times \underline{u} \quad (38)$$

which is equivalent to:

Designating the fuzing time of the missile as t_f , which is the time when the target intercepts the missile fuzing cone and is the beginning of the missile delay interval, T_f , defined as the period of time from fuzing to detonation. The target position at detonation, designated as the origin $\underline{0}$, can be determined by

$$\underline{0} = \underline{x}_T(t_f) + \underline{v}_T(t_f) \cdot T_f \quad (35)$$

where \underline{x}_T is the target position at t_f and \underline{v}_T is the target velocity at t_f , in the inertial coordinate system. The separation distance \underline{r} is the offset distance of the missile flight path projection from the origin. The vector \underline{P} , which is the line-of-sight vector from the missile at fuzing to the target at detonation is then given by:

$$\underline{P} = \underline{x}_m(t_f) - \underline{0} \quad (36)$$

where $\underline{x}_m(t_f)$ is the position of the missile at the fuzing time. Substituting Eq (35) into Eq (36) yields

$$\underline{P} = \underline{x}_m(t_f) - \underline{x}_T(t_f) - \underline{v}_T(t_f) \cdot T_f \quad (37)$$

The vector \underline{r} , which is the vector from the origin perpendicular to the missile flight line projection, is a function of \underline{P} ,

$$\underline{r} = (\underline{u} \times \underline{P}) \times \underline{u} \quad (38)$$

which is equivalent to:

$$\underline{r} = \underline{P} - (\underline{P} \cdot \underline{u}) \underline{u} \quad (39)$$

where \underline{u} is the missile unit velocity vector determined by the following:

$$\underline{u} = \frac{\underline{v}_m(t_f)}{\underline{v}_m(t_f)} \quad (40)$$

Combining Eqs (37), (39), and (40) yields:

$$\begin{aligned} \underline{r} = & \underline{\Delta X} - T_f \underline{v}_T - (\underline{\Delta X} \cdot \underline{v}_m) \underline{v}_m / \underline{v}_m^2 \\ & + (\underline{v}_T \cdot \underline{v}_m) \underline{v}_m T_f / \underline{v}_m^2 \end{aligned} \quad (41)$$

where all values are determined at the fusing time t_f , and where \cdot designates the vector dot product.

The ellipsoidal equation for iso- P_k contours in matrix form is:

$$R^2 = \underline{r}^T F \underline{r} \quad (42)$$

where \underline{r} , in Eq (41), can be expressed in the inertial coordinate system. F is most easily defined in the target coordinate system

$$F_T = \begin{bmatrix} 1/\sigma_x^2 & 0 & 0 \\ 0 & 1/\sigma_y^2 & 0 \\ 0 & 0 & 1/\sigma_z^2 \end{bmatrix} \quad (43)$$

where F is the scaling matrix determined to approximate the P_k data of detailed vulnerability studies by the ellipsoid

surfaces. In order to express F in the inertial coordinate system, the following similarity transformation must be made:

$$F = C_I^T F_T C_T^I \quad (44)$$

C_T^I is the transformation matrix from the inertial coordinate system to the target coordinate system; and C_I^T transforms from target to inertial coordinates. In determining C_T^I the order of rotation is through the heading angle (ψ_T), followed by flight path angle (γ_T), and then the bank angle (β_T), and finally through the angle of attack (α_T):

$$C_T^I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_T) & \sin(\alpha_T) \\ 0 & -\sin(\alpha_T) & \cos(\alpha_T) \end{bmatrix} \cdot \begin{bmatrix} \cos(\beta_T) & 0 & -\sin(\beta_T) \\ 0 & 1 & 0 \\ \sin(\beta_T) & 0 & \cos(\beta_T) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma_T) & \sin(\gamma_T) \\ 0 & -\sin(\gamma_T) & \cos(\gamma_T) \end{bmatrix} \cdot \begin{bmatrix} \cos(\psi_T) & -\sin(\psi_T) & 0 \\ \sin(\psi_T) & \cos(\psi_T) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (45)$$

Therefore, the transformation from the target coordinate system to the inertial coordinate system is given by:

$$C_I^T = (C_T^I)^T \quad (46)$$

Finally, the terminal cost, defined as $F(X(t_f); t_f)$ is given by:

$$F(X(t_f); t_f) = e^{-R^2} \quad (47)$$

where R^2 is defined by Eq (42).

The type missile used for this simulation is an air-to-air heat seeking missile. The tail of the aircraft was chosen as the aim point (or origin) for the target. Since the missile is guided towards the tail of the evading aircraft, kill probabilities are higher if the detonation is forward rather than behind the target aim point. A bias term is used to shift the center of the ellipsoids forward to account for more probable fusing forward of the aim point. The amount of shift is designated as \underline{b} , then the vector from the ellipsoid center to the tip of the \underline{r} vector is

$$\underline{D} = \underline{r} - \underline{b} \quad (48)$$

where \underline{r} is the "closest approach point" of the projected missile flight line projection to the aim point (the P_k analysis was based on r as a parameter). The missile used for this algorithm has a shift vector \underline{b} of:

$$\begin{aligned} b_x &= 0 \\ b_y &= 6 \text{ ft} \\ b_z &= 2 \text{ ft} \end{aligned} \quad (49)$$

and, therefore, the ellipsoid equation for the iso- P_k contours in matrix form becomes:

$$R^2 = \underline{D}^T F \underline{D} \quad (50)$$

For the missile used, the weighting matrix, F_T , in the target coordinate system is:

$$F_T = \begin{bmatrix} \frac{1}{(21)^2} & 0 & 0 \\ 0 & \frac{1}{(22)^2} & 0 \\ 0 & 0 & \frac{1}{(16)^2} \end{bmatrix} \quad (51)$$

Therefore, the terminal cost for the missile used in this program becomes:

$$F(x(t_f), t_f) = e^{-\underline{D}^T C_I^T \begin{bmatrix} \frac{1}{(21)^2} & 0 & 0 \\ 0 & \frac{1}{(22)^2} & 0 \\ 0 & 0 & \frac{1}{(16)^2} \end{bmatrix} C_T^I \underline{D}} \quad (52)$$

IV. Terminal Constraint

A typical air-to-air missile has a fuzing cone angle (FCA) of approximately 60 degrees. The proportional navigation steering attempts to maneuver the missile so that the target is constrained inside the fuzing cone angle at all times. During the terminal portion of the flight, as the missile approaches the target, practical limitations on missile maneuvering will allow the target to reach the fuzing cone angle. At the time when the line-of-sight from the missile to the target lies on the missile fuzing cone the missile fuzing delay is initiated, and, following a preset delay time, detonation is programmed to occur. The fuzing cone angle and delay times are normally chosen to give "good" fragment patterns.

The terminal constraint for this problem is, therefore, when the line-of-sight from the missile to the target equals the fuzing cone angle. Refer to Fig. 7 for a schematic of the terminal constraint.

Let A equal the unit missile axis vector and S equal the line-of-sight (LOS) vector from the missile to the target; therefore, when the LOS lies on the edge of the fuzing cone angle then

$$\mathbf{S} \times \mathbf{A} = |\mathbf{S}| |\mathbf{A}| \sin(\text{FCA}) \quad (53)$$

where $|\mathbf{A}|$ is equal to 1. S which is the LOS vector from the missile to the target is then equal to

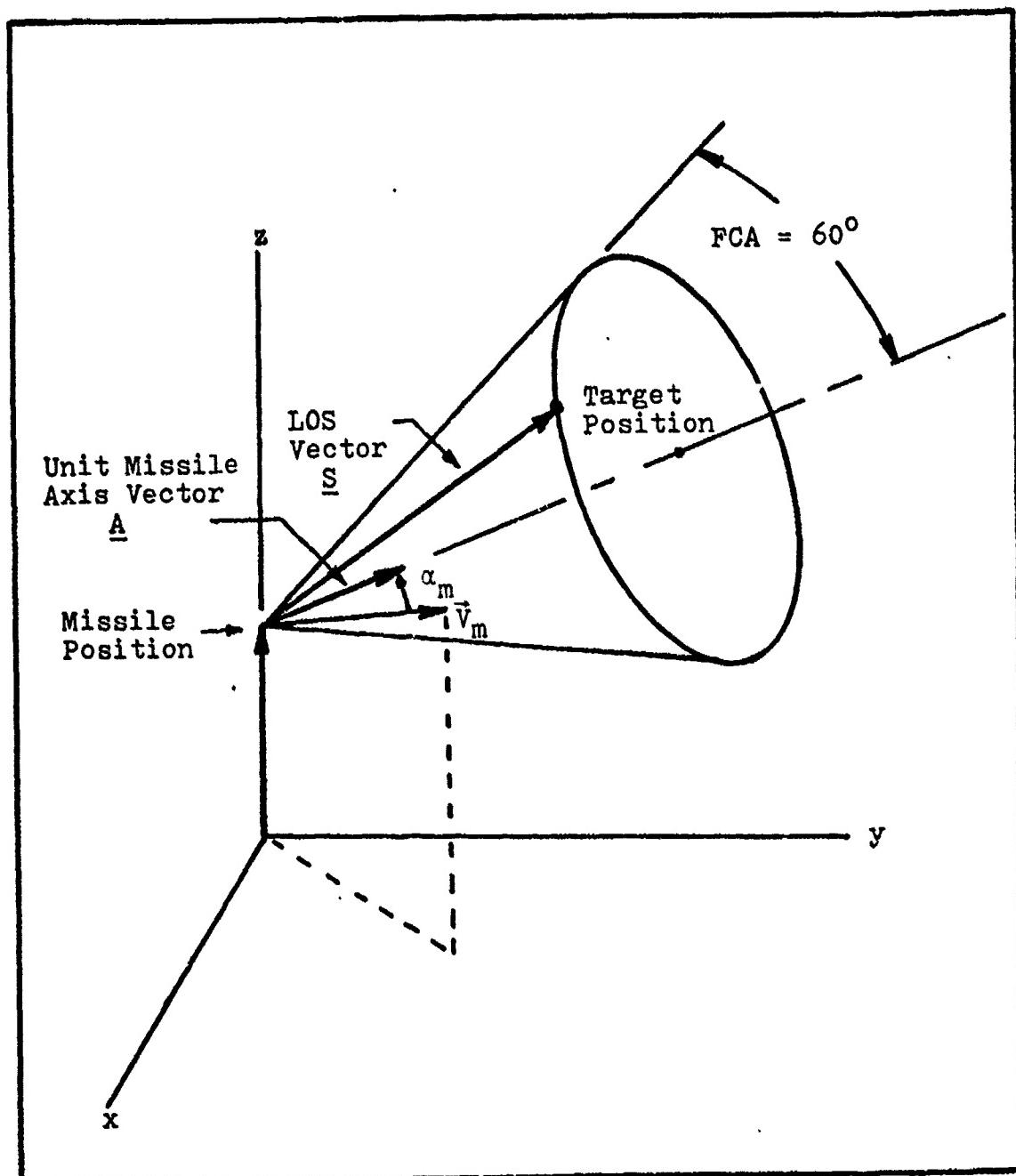


Fig. 7. Terminal Constraint Geometry

$$\underline{S} = \begin{bmatrix} X_T - X_m \\ Y_T - Y_m \\ Z_T - Z_m \end{bmatrix} \quad (54)$$

The terminal constraint, designated as $\psi(X(t_f), t_f)$, is then given as:

$$\psi(X(t_f), t_f) = |\underline{S} \times \underline{A}|^2 - |\underline{S}|^2 \sin^2(\text{FCA}) \quad (55)$$

\underline{A} which is the unit missile axis vector in the missile coordinate system is equal to

$$\underline{A} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (56)$$

In order to obtain \underline{A} in the inertial coordinate system the transformation from the missile coordinates to the inertial coordinate system must be made. The transformation from the missile to the inertial coordinate system, A_I^m , is of the same form as the transformation matrix for the target A_I^T , and recalling that

$$A_I^m = (A_m^I)^T \quad (57)$$

then A_I^m becomes

$$A_I^m = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_m) & \sin(\alpha_m) \\ 0 & -\sin(\alpha_m) & \cos(\alpha_m) \end{bmatrix} \cdot \begin{bmatrix} \cos(\beta_m) & 0 & -\sin(\beta_m) \\ 0 & 1 & 0 \\ \sin(\beta_m) & 0 & \cos(\beta_m) \end{bmatrix} \cdot$$
(58)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma_m) & \sin(\gamma_m) \\ 0 & -\sin(\gamma_m) & \cos(\gamma_m) \end{bmatrix} \cdot \begin{bmatrix} \cos(\psi_m) & -\sin(\psi_m) & 0 \\ \sin(\psi_m) & \cos(\psi_m) & 0 \\ 0 & 0 & 1 \end{bmatrix}^T$$

where the order of rotation is the angle of attack, then bank angle, followed by flight path angle, and finally by a heading change. Finally, \underline{A} in the inertial coordinate system is

$$\underline{A} = A_I^m \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$
(59)

The terminal constraint equation is then defined by Eq (55) where \underline{S} and \underline{A} are defined by Eqs (54) and (59) respectively, and FCA depends on the missile being used. For the missile used in this program the fuzing cone angle is 60 degrees.

V. Differential Dynamic Programming Equations

The second order algorithm for fixed end point problems with the final time (t_f) given implicitly is discussed in Ref (1:38). The derivation of the required equations will not be given here, but the equations to be used for this particular problem will be discussed.

Initially, a first order algorithm was considered for use in this program; therefore, the following sample problem was attempted with the first order method:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 - x_1^3 + u\end{aligned}\tag{60}$$

The rate of convergence on this nonlinear problem using the first order method was very slow. The second order algorithm was then attempted on the same problem with the rate of convergence being greatly improved over that of the first order method. The optimal missile evasion problem is quite nonlinear and as previously outlined has a 12 dimensional state vector; therefore, the second order algorithm was selected even though of the disadvantage of taking a large number of second order derivatives.

The second order method consists of two phases; the first phase, optimization phase, uses backward integration of the costate equations to determine the sensitivities of the cost function for proper adjustment of the controls. Throughout this phase the value of the terminal constraint function is allowed to wander. The second

phase, which is designated as the restoration phase, starts with controls obtained in the optimization phase and then attempts to restore the constraint.

In general, the function to be minimized can be given in the general form of:

$$\begin{aligned} J(\underline{x}_0, b, t_f; t_0) = & \int_{t_0}^{t_f} L(\underline{x}, \underline{u}, t) dt \\ & + F(\underline{x}(t_f); t_f) \\ & + b \psi(\underline{x}(t_f); t_f) \end{aligned} \quad (61)$$

where the final time t_f is given implicitly. For this particular problem, the controls are indirectly constrained by the integral cost term

$$L(\underline{x}, \underline{u}, t) = 1/2(RA \cdot (\alpha_{T_c})^2 + RB \cdot (\beta_T)^2 + RC \cdot (C_{T_c})^2) \quad (62)$$

where RA, RB, and RC, the weights put on the controls, are selected as the inverse of the maximum value expected for that control. For this problem the values are

$$\begin{aligned} RA &= 10 \\ RB &= .1 \\ RC &= 73 \end{aligned} \quad (63)$$

$F(\underline{x}(t_f), t_f)$, the terminal cost, is defined by Eq (52), $\psi(\underline{x}(t_f), t_f)$, the terminal constraint, is defined by Eq (55), b is the time invariant Lagrange multiplier, and the state variable differential equations are of the form $\dot{x} = f(x, u, t)$.

The necessary sensitivity equations to be integrated in order to minimize Eq (61) are

$$\begin{aligned}
 -\dot{a} &= H - H(\bar{x}, \bar{u}, v_x; t) \\
 -\dot{v}_x &= H_x + v_{xx}(f - f(\bar{x}, \bar{u}; t)) \\
 -\dot{v}_{xb} &= (f_x + f_u s_1)^T v_{xb} \\
 -\dot{v}_{xt_f} &= (f_x + f_u s_1)^T v_{xt_f} \\
 -\dot{v}_{bt_f} &= -v_{xb}^T f_u H_{uu}^{-1} f_u^T v_{xt_f} \\
 -\dot{v}_{bb} &= -v_{xb}^T f_u H_{uu}^{-1} f_u^T v_{xb} \\
 -\dot{v}_{xx} &= H_{xx} + f_x^T v_{xx} + v_{xx} f_x \\
 &\quad - (H_{ux} + f_u^T v_{xx})^T H_{uu}^{-1} (H_{ux} + f_u^T v_{xx}) \\
 -\dot{v}_{t_f t_f} &= -v_{xt_f}^T f_u H_{uu}^{-1} f_u^T v_{xt_f} \tag{64}
 \end{aligned}$$

where all quantities are evaluated at \bar{x} , \bar{u} , \bar{t}_f and u^* unless otherwise specified. The terms designated with the bar above indicate the values along the nominal trajectory for which that variable is used. The optimized control for the next iteration is given by

$$\begin{aligned}
 u(t) = u^*(t) &+ s_1(t) dx(t) + s_2(t) db \\
 &+ s_3(t) dt_f \tag{65}
 \end{aligned}$$

where $s_1(t)$, $s_2(t)$ and $s_3(t)$ are given as:

$$\begin{aligned}
 s_1(t) &= -H_{uu}^{-1} (H_{ux} + f_u^T v_{xx}) \\
 s_2(t) &= -H_{uu}^{-1} f_u^T v_{xb}
 \end{aligned}$$

$$s_3(t) = -H_{uu}^{-1} f_u^T v_{xt_f} \quad (66)$$

The functions with the subscripts indicate the partial of that function with respect to that subscript. Note that db and dt_f are zero except during the restoration phase, and the third, fourth, fifth, sixth, and eighth equations of (64) are used only during the restoration phase.

The Hamiltonian, H , is defined as

$$H = L + v_x^T f \quad (67)$$

The necessary condition for an optimum control, \underline{u}^* , is that $H_u = 0$. Taking the partial derivative of the Hamiltonian with respect to u yields the following \underline{u}^* :

$$\begin{aligned} u_1^* &= -v_{11}/\tau_p * RA \\ u_3^* &= -v_{12}/\tau_T * RC \end{aligned} \quad (68)$$

where v_{11} is defined as the eleventh component of the costate equation in the \dot{v}_x equation (64), etc. Since u_2 is an argument of the sin and cos, the u_2^* equation is transcendental. A root finding subroutine is used to solve for u_2^* from the following transcendental equation:

$$\begin{aligned} & RB(u_2^*) + \cos(u_2^*)(v_8 1/2 \rho_o e^{-(z_T/z_o)} s_T \\ & \cdot v_T CL_{aT} \alpha_T/m_T \cos(\gamma_T)) + \sin(u_2^*) \\ & \cdot (-v_{10} 1/2 \rho_o e^{-(z_T/z_o)} s_T v_T CL_{aT} \alpha_T/m_T) = 0 \end{aligned} \quad (69)$$

The boundary conditions required for the dif-

ferential equations (64) are:

$$a(t_f) = 0$$

$$v_x = F_x + \psi_x^T \bar{b}$$

$$v_{xb} = \psi_x^T$$

$$v_{xt_f} = H_x + v_{xx} f$$

$$v_{bt_f} = \psi_x f$$

$$v_{bb} = 0$$

$$v_{xx} = F_{xx} + \bar{b} \psi_{xx}$$

$$v_{t_f t_f} = \langle H_x, f \rangle + \langle f, v_{xx} f \rangle \quad (70)$$

where $\langle \rangle$ signifies the inner product, and \bar{b} is the nominal Lagrange multiplier. The computational procedure for the program is outlined in the next chapter.

VI. Computational Procedure

In order to understand more clearly the computational procedure used for the optimal missile evasion differential dynamic programming model a synopsis of the program will now be discussed. As was stated earlier in this thesis the main concern of this program is to determine the minimum P_k of the missile, as well as the optimum controls used by the aircraft. The P_k obtained is the minimum for a particular launch condition when the evading target uses optimum controls: angle of attack, bank angle, and coefficient of thrust. In the real world situation the P_k , for the same launch condition, would be equal to or greater than that P_k obtained by this program depending on how skillful the evading target pilot was.

For this program the terminal condition is known, that is, the stopping criteria occurs when the LOS from the missile to the target intercepts the missile's fuzing cone as discussed in Chapter IV. The duration over which the control is to be applied, however, is not fixed. The interval $[t_0, t_f]$ is, therefore, not specified explicitly. An initial time interval of one second was selected to test this program. A nominal control, \bar{u} , is then loaded into an array for the three controls. Each control is selected as a constant throughout the interval using a step size of .1 seconds, with the nominal angle of attack, \bar{u}_1 , of zero degrees, the nominal bank angle, \bar{u}_2 , of zero degrees, and the nominal coefficient of thrust, \bar{u}_3 , of .025. The nominal value of

the coefficient of thrust is based upon a 55,000 lb airplane at 20,000 ft using military power (7000 lbs of thrust).

A brief outline of the method used in the algorithm is depicted in Fig. 8; and a more detailed analysis of the program is described in steps 1 through 6.

Step 1 - Backward Integration of the State Equations Using Nominal Control

Since it is known approximately where the terminal conditions occur, these terminal conditions are used to initialize the state equations in order to integrate the states backwards in time the one second interval. Referring to Fig. 9 for a geometrical interpretation of the terminal conditions selected for this problem, the initialization of the state equations for the backwards integration are:

$$\begin{aligned} X_m - X_T &= 5 \text{ ft} \\ Y_m - Y_T &= -8.66 \text{ ft} \\ Z_m &= 20,000 \text{ ft} \\ Z_T &= 20,000 \text{ ft} \\ V_m &= 1866.4 \text{ ft/sec} \\ V_T &= 829.5 \text{ ft/sec} \\ \psi_m &= 30 \text{ deg} \\ \psi_T &= 0 \text{ deg} \\ \gamma_m &= 0 \text{ deg} \\ \gamma_T &= 0 \text{ deg} \\ \alpha_T &= 0 \text{ deg} \\ C_T &= .025 \end{aligned} \tag{71}$$

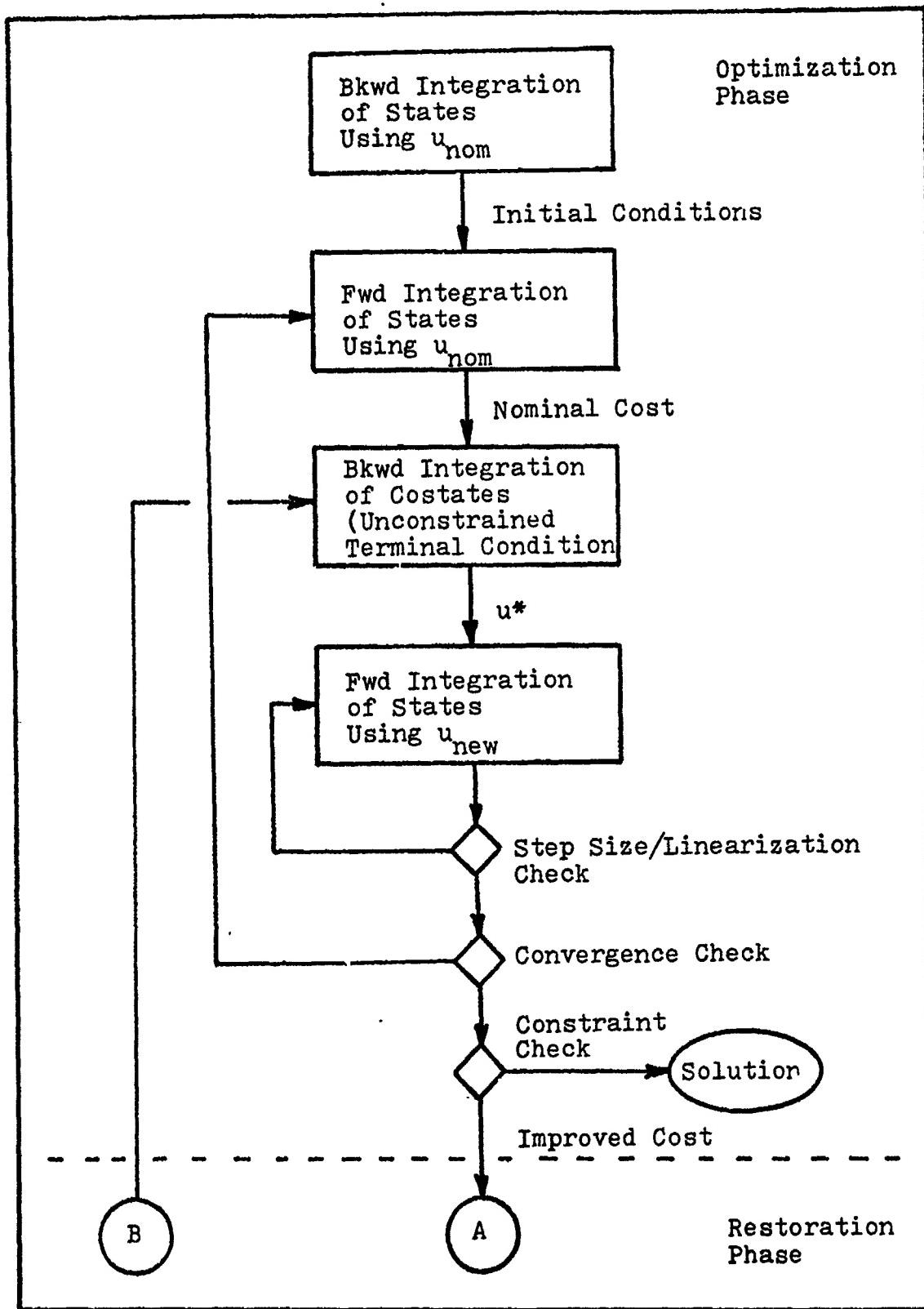


Fig. 8. Differential Dynamic Programming Model Flow Chart

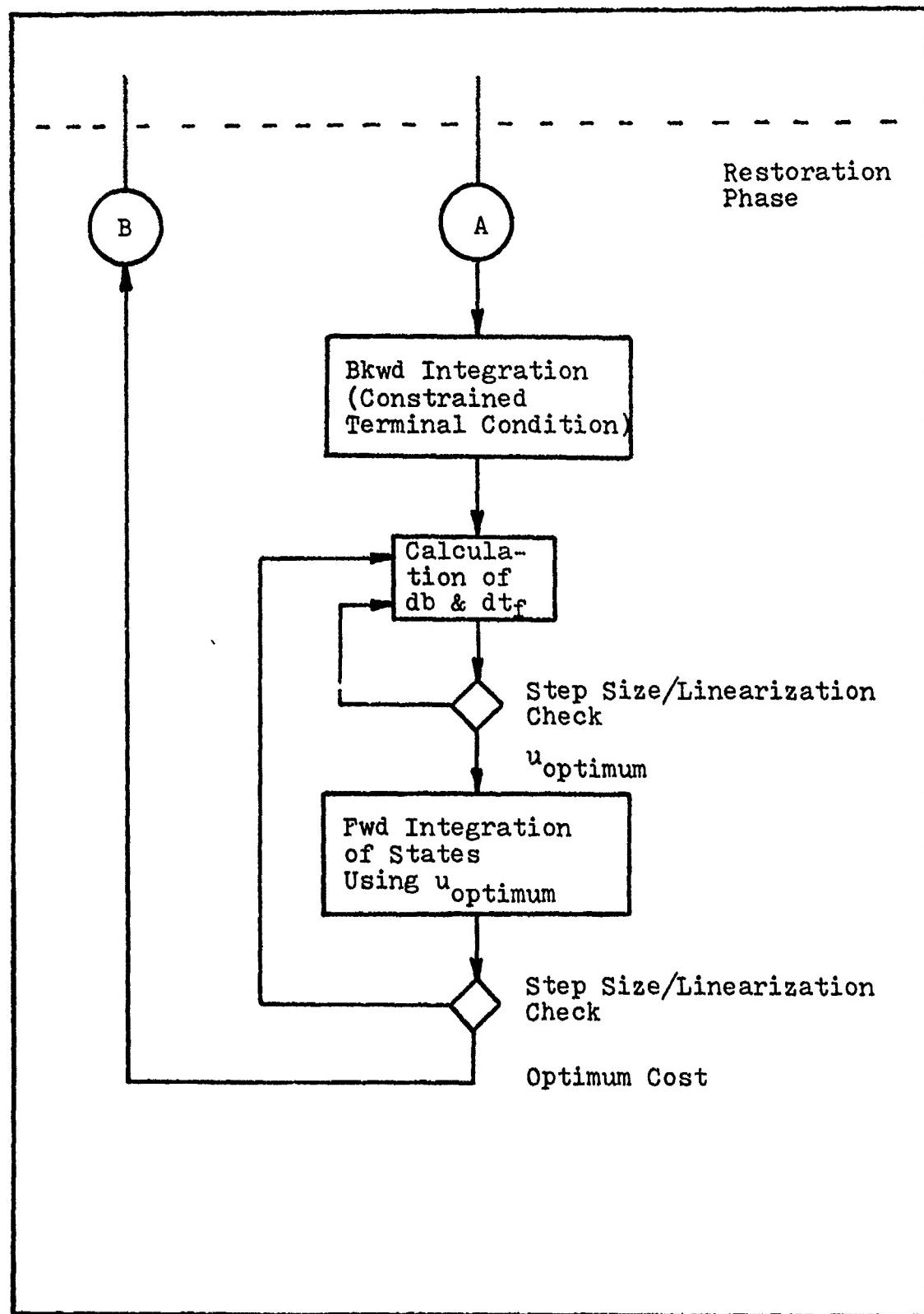


Fig. 8. Differential Dynamic Programming Model Flow Chart (Concluded)

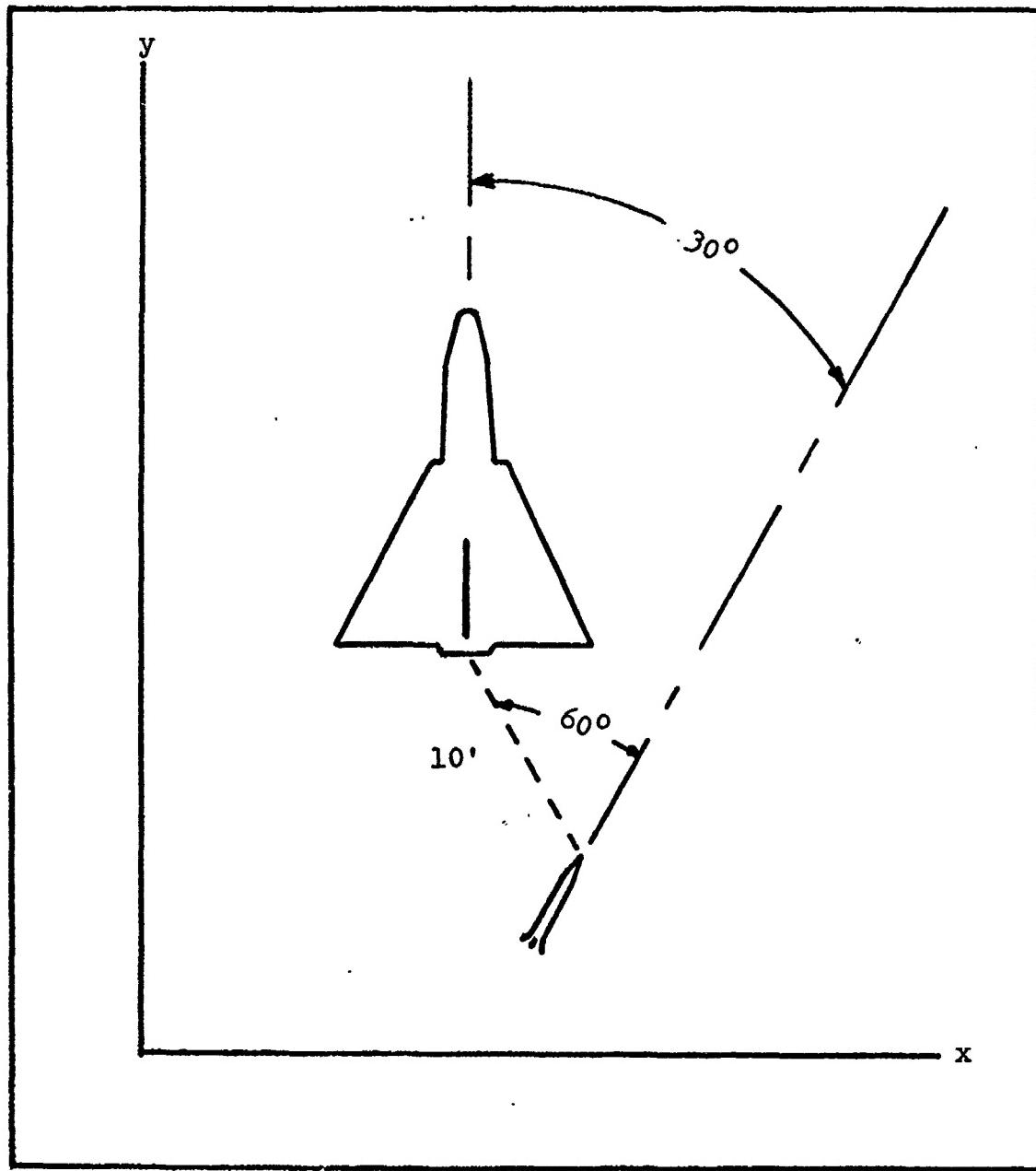


Fig. 9. Initialization of Terminal Constraint

The state equations are then integrated backward from t_f to t_0 using the nominal controls and using the above starting conditions. The backward integration is stopped at t_0 where values for the states are obtained and designated as $\underline{X}(t_0)$

Step 2 - Forward Integration of the State Equations

The state equations are integrated from t_0 to t_f using the same nominal controls and using the initial conditions $\underline{X}(t_0)$ as determined from step 1. During this forward integration the nominal cost is determined using

$$J_{\text{nom}} = F(X(t_f); t_f) + \bar{\psi} \psi(X(t_f); t_f) \\ + 1/2 \int_{t_0}^{t_f} (RA \cdot (u_1)^2 + RB \cdot (u_2)^2 + RC \cdot (u_3)^2) dt \quad (72)$$

where $F(X(t_f); t_f)$ is the terminal cost as discussed in Chapter III, $\psi(X(t_f); t_f)$ is the terminal constraint as discussed in Chapter IV and $\bar{\psi}$ is the nominal Lagrange multiplier which was set equal to 1. The procedure used for evaluating the terminal cost and constraint is discussed in step 3. The states are then evaluated at t_f and designated as $\underline{X}(t_f)$.

Step 3 - Backward Integration with Unconstrained Terminal Condition

The following set of differential equations are integrated backwards from t_f to t_0 :

$$\begin{aligned} \text{States : } & X \\ & \dot{A} \end{aligned}$$

$$\begin{aligned} \dot{v}_x \\ \dot{v}_{xx} \end{aligned} \quad (73)$$

The starting conditions for integrating the above equations backwards are

$$\begin{aligned} \underline{x} &= \underline{x}(t_f) \\ A &= 0 \\ v_x &= F_x + \psi_x^T \bar{b} \\ v_{xx} &= F_{xx} + \bar{b} \psi_{xx} \end{aligned} \quad (74)$$

The terminal conditions for the costate (\dot{v}_x) equations involve taking the first partial derivative of the terminal cost (F_x) and the terminal constraint (ψ_x); while the terminal conditions for the \dot{v}_{xx} equations involve taking the second partial derivative of the cost (F_{xx}) and constraint (ψ_{xx}). Both the terminal cost and terminal constraint are functions of a large number of the states, states one through eleven for the terminal cost, and states one through ten for the terminal constraint. Since both the first and second partial derivatives of these functions are required, a large number of terms are involved. Since high accuracy is not required for the values used for the initialization of the differential Equations (74), the analytic solutions to the initialization of the \dot{v}_x and \dot{v}_{xx} equations will not be attempted. The derivative v_x is approximated using divided differences

$$F_{x_i} = \frac{F(x_0 + \Delta x_i) - F(x_0 - \Delta x_i)}{2\Delta x_i} \quad i = 1, 2, \dots, 11$$

and

$$\psi_{x_i} = \frac{\psi(x_0 + \Delta x_i) - F(x_0 - \Delta x_i)}{2\Delta x_i} \quad i = 1, 2, \dots, 10 \quad (75)$$

where F_{x_i} is the numerical value of the first partial derivative of the terminal cost with respect to the i^{th} state; and ψ_{x_i} is the numerical value of the first partial derivative of the terminal constraint with respect to the i^{th} state. The nominal vector, x_0 , is the value of the states at the final time. Δx_i is set equal to 1% of the value of that state at the final time plus an epsilon. The epsilon is added in order to provide a usable perturbation even when the nominal value is small or zero. This prevents division by zero. The particular values for the Δx_i are:

$$\begin{aligned}\Delta x_1 &= .01 x_1(t_f) + 1 \text{ ft} \\ \Delta x_2 &= .01 x_2(t_f) + 1 \text{ ft} \\ \Delta x_3 &= .01 x_3(t_f) + 1 \text{ ft} \\ \Delta x_4 &= .01 x_4(t_f) + 1 \text{ ft} \\ \Delta x_5 &= .01 x_5(t_f) + 1 \text{ ft/sec} \\ \Delta x_6 &= .01 x_6(t_f) + 1 \text{ ft/sec} \\ \Delta x_7 &= .01 x_7(t_f) + 1 \text{ degree} \\ \Delta x_8 &= .01 x_8(t_f) + 1 \text{ degree} \\ \Delta x_9 &= .01 x_9(t_f) + 1/2 \text{ degree} \\ \Delta x_{10} &= .01 x_{10}(t_f) + 1/2 \text{ degree} \\ \Delta x_{11} &= .01 x_{11}(t_f) + 1/2 \text{ degree} \end{aligned} \quad (76)$$

Subroutine FFX calculates F_{x_i} , and subroutine FPX calculates

ψ_{x_i} .

The derivative V_{xx} is approximated using

$$F_{x_i x_j} = \frac{[F(X_0 + \Delta X_i + \Delta X_j) - F(X_0 - \Delta X_i + \Delta X_j) - F(X_0 + \Delta X_i - \Delta X_j) + F(X_0 - \Delta X_i - \Delta X_j)]}{4 \Delta X_i \Delta X_j}$$

for $i \neq j$ $i = 1, 2, \dots, 11$

$j = 1, 2, \dots, 11$

and

$$F_{x_i x_j} = \frac{F(X_0 + \Delta X_i) - 2F(X_0) + F(X_0 - \Delta X_i)}{\Delta X_i^2} \quad (77)$$

for $i = j$ $i = 1, 2, \dots, 11$

where $F_{x_i x_j}$ is the numerical value of the second partial derivative of the terminal cost. $\psi_{x_i x_j}$, the second partial derivative of the terminal constraint, is computed in the same way. The ΔX_i and ΔX_j are the same values as that previously determined in Eq (76). Subroutine FFXX calculates $F_{x_i x_j}$ and subroutine FPXX calculates $\psi_{x_i x_j}$.

As previously stated in step 2, at the end of the forward integration the terminal cost and terminal constraint values are required. For the terminal cost, subroutine FFXX is used to calculate this value. The terminal cost is calculated in that portion of the routine where zero displacement of the states is required. The terminal constraint is calculated by subroutine FPXX using the same procedures.

Returning to the main program with the boundary conditions, Equations (73) are integrated backward from t_f to t_0 . Subroutine F is used for this backward integration. During this backward integration H is minimized with respect to u to obtain \hat{u}^* . As previously stated, the equation for u_2 is transcendental, therefore, subroutine F determines u_2^* from the transcendental equation at each step. The solution for u_2^* is obtained by iteration using the root finding approximation

$$f(u_{2_0}) + f'(u_{2_0}) \delta u_2 \approx 0 \quad (78)$$

or solving for δu_2

$$\delta u_2 = - \frac{f(u_{2_0})}{f'(u_{2_0})} \quad \text{if } |f'(u_{2_0})| > 0 \quad (79)$$

where δu_2 is the change in u_2 from one iteration to the next and $f(u_{2_0})$ is the transcendental function evaluated at u_{2_0} . $f'(u_{2_0})$ is the derivative with respect to u_2 evaluated at u_{2_0} . The new updated estimate is then

$$u_{2_0} + \delta u_2 \quad (80)$$

This iterative type procedure continues until the solution converges (i.e., δu_2 becomes less than a specified tolerance). If $|f'(u_{2_0})|$ approximately equals zero then the second derivative has to be taken and δu_2 becomes

$$\delta u_2 = \pm \sqrt{-\frac{2 f(u_{2_0})}{f''(u_{2_0})}} \quad (81)$$

where the proper sign is chosen such that $|u_{2_0} + \delta u_2|$ is minimized.

The \dot{v}_x and \dot{v}_{xx} equations are integrated backwards using the values of u^* . The \dot{A} equation is integrated backward using the difference between u^* and u_{nominal} . A test parameter value of .0005 was preselected for A . Throughout the entire backward integration the values of $s_1(t)$, which is defined by Eq (66), are stored. The time is noted when A exceeds the preset value of .0005, and is designated as NEFF(2), the time of an "effective" change in A .

Step 4 - Forward Integration of State Equations Using Improved Control

The state equations are now integrated forward again from t_0 to t_f maintaining the same initial conditions but using the new control

$$u_{\text{new}} = u^* + s_1(t) dx(t) \quad (82)$$

where $s_1(t)$ was calculated in step 3 and $dx(t)$ is the difference between the states using the nominal control and the states using the u_{new} control designated x_n . During the first iteration, $dx(t_0)$ equals zero and then dx is calculated for the remaining steps to t_f . Throughout the entire forward integration the new cost, designated as J_{star} , is

calculated. J_{star} includes the integral cost, the terminal cost, and the terminal constraint penalty calculated the same way as in step 2; but all functions are evaluated using $\underline{u}_{\text{new}}$. The requirement for a satisfactory new control is that the change in cost be greater than some constant times $A(t_0)$. Jacobson and Mayne Ref (1:25) suggest initially that the constant equal .5; therefore, the requirement for a valid new control, $\underline{u}_{\text{new}}$, is

$$J_{\text{nom}} - J_{\text{star}} \geq C A(t_0) \quad (83)$$

If inequality (83) is satisfied then the $\underline{u}_{\text{new}}$ control is loaded into the $\underline{u}_{\text{nominal}}$ control and then step 2 is repeated. If inequality (83) is not satisfied then a step size adjustment method is required to prevent overstepping the region of linearity. The step size adjustment method consists of determining the position on the integration interval half way between NEFF(2), which is the first point on the backward integration where A exceeded .0005, and t_0 . This new position is called MK in the main program. The $\underline{u}_{\text{new}}$ control is then changed by the following method:

$$\underline{u}_{\text{new}} = \underline{u}_{\text{nominal}}$$

on the interval from t_0 to MK and then

$$\underline{u}_{\text{new}} = \underline{u}^* + s_1(t) dx(t) \quad (84)$$

on the interval from MK to t_f . Using the new control, step

\downarrow is then repeated but now the change in states, $dx(t)$, is zero from t_0 to MK since the same control is used over that interval, and on the interval from MK to t_f , $dx(t)$ is calculated as before. Again the change in cost is evaluated and the criteria for a valid new control is

$$J_{\text{nom}} - J_{\text{star}} \geq C A(MK) \quad (85)$$

The step size adjustment method criteria specifies that the change in cost must be greater than a predetermined value. If the criteria is not met, MK must be moved toward the final time so the old control is used over more of the total interval. This iterative type procedure is continued until MK and NEFF(2) coincide. When this condition is satisfied then C is set equal to zero. When C is set equal to zero, the most optimum trajectory has been found using C equal to .5. The u_{new} control is then again loaded into the u_{nominal} control and step 2 is repeated. The new criteria for a valid control is that

$$J_{\text{nom}} - J_{\text{star}} \geq 0 \quad (86)$$

This is a more refined criteria in that as long as J_{star} is less than J_{nom} the control is valid. Again this iterative type procedure is used until MK and NEFF(2) coincide. The most optimum control has been determined by treating the problem as a free end point, that is the terminal constraint cri-

teria has not been satisfied. The next step will be to restore the terminal constraint condition, Eq (55).

Step 5 - Backward Integration with Constrained Terminal Condition.

The following set of differential equations are integrated backwards from t_f to t_0 :

States : \dot{x}

$$\begin{aligned}\dot{v}_x \\ \dot{v}_{xx} \\ \dot{v}_{xb} \\ \dot{v}_{bb} \\ \dot{v}_{xt_f} \\ \dot{v}_{bt_f} \\ \dot{v}_{t_f t_f}\end{aligned}$$

(87)

with the initial conditions of:

$$x = x_n(t_f)$$

$$v_x = F_x + \psi_x^T \bar{\delta}$$

$$v_{xx} = F_{xx} + \bar{\delta} \psi_{xx}$$

$$v_{xb} = \psi_x^T$$

$$v_{bb} = 0.$$

$$v_{xt_f} = H_x + v_{xx} f$$

$$v_{bt_f} = \psi_x f$$

$$v_{t_f t_f} = \langle H_x, f \rangle + \langle f, v_{xx} f \rangle \quad (88)$$

The initial conditions for the \dot{v}_x , \dot{v}_{xx} and \dot{v}_{xb} equations are determined the same way as in step 3. The initial conditions for the \dot{v}_{xt_f} , \dot{v}_{bt_f} and $\dot{v}_{t_f t_f}$ are calculated analytically by subroutine STCST. During the backward integration using u_{new} , $\beta_2(t)$ and $\beta_3(t)$ are determined from t_f to t_o where $\beta_2(t)$ and $\beta_3(t)$ are defined by:

$$\begin{aligned}\beta_2(t) &= -H_{uu}^{-1} f_u^T v_{xb} \\ \beta_3(t) &= -H_{uu}^{-1} f_u^T v_{xt_f}\end{aligned} \quad (66)$$

The new control to be evaluated on the next forward integration is given by

$$u(t) = u_{new} + \beta_2(t) db + \beta_3(t) dt_f \quad (89)$$

where u_{new} is defined in Eq (84) and where db and dt_f are given by

$$\begin{bmatrix} db \\ dt_f \end{bmatrix} = -E \begin{bmatrix} v_{bb} & v_{bt_f} \\ v_{t_f b} & v_{t_f t_f} \end{bmatrix}^{-1} \begin{bmatrix} v_b \\ v_{t_f} \end{bmatrix} \quad (90)$$

All values are determined at t_o , therefore, db and dt_f are constants for a particular integration cycle. E is also a constant for a particular integration cycle which initially equals 1 as suggested in Ref (1:43). The quantity dt_f is the

change in the final time of the trajectory; and for this problem was limited to a maximum of 10% of the final time. That is, the criteria for dt_f is

$$|dt_f| \leq .1 t_f \quad (91)$$

where t_f is the final time of the trajectory. If inequality (91) is not satisfied then E is set to E/2 and then db and dt_f are re-evaluated by Eq (90). If inequality (91) is satisfied then proceed on to step 6.

Step 6 - Forward Integration of State Equations Using Optimum Control

The control given by Eq (89) is designated as u_{opt} in the program. The state equations are then integrated from t_0 to t_f using u_{opt} . The values of the states are determined and the terminal constraint is then calculated by using subroutine FPXX. The value of this optimum constraint is designated as PCOSTP, while the constraint evaluated in step 4 was designated as PCOSTN. In order to determine if there has been an improvement in the end point error the following criteria must be satisfied:

$$|PCOSTN|^* > |PCOSTP| \quad (92)$$

where again PCOSTN is the value of the terminal constraint by using u_{new} , and PCOSTP is the value of the terminal constraint using u_{opt} . If inequality (92) is not satisfied then set E to E/2 and return to Eq (90) where db and dt_f are re-

evaluated. If inequality (92) is satisfied then the \underline{u}_{opt} control is loaded into the $\underline{u}_{nominal}$ control and step 3 is repeated. The stopping criteria for this restoration phase is when the terminal constraint satisfies

$$|\psi(\mathbf{x}(t_f), t_f)| \leq .005 \quad (93)$$

The final values obtained by this program are the following:

1. The trajectory of the missile,
2. The optimum trajectory of the evading target,
3. The optimum controls used by the target and,
4. The value of the terminal cost, which is the minimum P_k the missile would obtain for the initial conditions as outlined by Eq (71).

A detailed schematic of the differential dynamic algorithm used for this program is illustrated in Fig. 10, Appendix A.

VII. Results

As previously stated this study attempted to determine the optimal controls to be used by an aircraft evading an air-to-air missile; with the controls consisting of the angle of attack, the bank angle, and the coefficient of thrust. The state vector for this missile evasion problem included the distance components between the target and missile, the velocity of both target and missile, the heading of target and missile, the flight path angle of target and missile, the angle of attack of the target, and the coefficient of thrust of the target. The performance of the system was measured by the minimization of the terminal cost function, which was the P_k of the missile.

The objective of the terminal cost function was to make the model correspond to detailed simulation of the end game which incorporated the shape, orientation and vulnerability of the aircraft. The individual vulnerability of a particular aircraft was not considered, but only the generic character. The P_k ellipsoid parameters were chosen to fit detailed simulation results reasonably well. Previous optimal missile evasion investigations modeled the terminal cost function as the magnitude of miss distance only and did not consider the vulnerability of the aircraft as a function of the orientation geometry.

Due to the complexity of the algorithm for this optimization program and the imposed time limit, the desired optimal

solutions for this investigation were not obtained.

Prior to attempting this optimal missile evasion problem the following system consisting of two states was solved using both the first order and second order methods,

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 - x_1^3 + u\end{aligned}\quad (60)$$

with the terminal constraint consisting of

$$\psi(x(t_f); t_f) = x_1 - 2x_2 - 2 \quad (94)$$

and the integral cost consisting of

$$L = 1/2 \int_0^5 (x_1^2 + x_2^2 + u^2) dt \quad (95)$$

Both the first order and second order techniques were attempted in order to evaluate the feasibility of using either of these techniques for the optimal missile evasion problem, which consists of a system with 12 state variables. In this sample problem it was determined that the rate of convergence using the first order technique was extremely slow. With this in mind it was determined that if the first order method was attempted on the evasion problem then the slow rate of convergence would make it extremely difficult to obtain the many desired solutions, thus the second order technique was selected as the best candidate method for this program.

The greatest disadvantage of using the second order technique is the high dimension of the system of differential equations. For any optimization problem with an unspecified final time, using the technique outlined in Ref (1:46), the following equations must be calculated:

$$\begin{aligned}
 -\dot{a} &= H - H(\bar{x}, \bar{u}, v_x; t) \\
 -\dot{v}_x &= H_x + V_{xx}(f - f(\bar{x}, \bar{u}; t)) \\
 -\dot{v}_{xb} &= (f_x + f_u \beta_1)^T v_{xb} \\
 -\dot{v}_{xt_f} &= (f_x + f_u \beta_1)^T v_{xt_f} \\
 -\dot{v}_{bt_f} &= -V_{xb}^T f_u H_{uu}^{-1} f_u^T v_{xt_f} \\
 -\dot{v}_{bb} &= -V_{xb}^T f_u H_{uu}^{-1} f_u^T v_{xb} \\
 -\dot{v}_{xx} &= H_{xx} + f_x^T v_{xx} + V_{xx} f_x \\
 &\quad - (H_{ux} + f_u^T v_{xx})^T H_{uu}^{-1} (H_{ux} + f_u^T v_{xx}) \\
 -\dot{v}_{t_f t_f} &= -V_{xt_f}^T f_u H_{uu}^{-1} f_u^T v_{xt_f} \tag{64}
 \end{aligned}$$

with the control determined by:

$$\begin{aligned}
 u(t) = u^*(t) &+ \beta_1(t) dx(t) + \beta_2(t) db \\
 &+ \beta_3(t) dt_f \tag{65}
 \end{aligned}$$

and where β_1 , β_2 , and β_3 are calculated by:

$$\begin{aligned}
 \beta_1(t) &= -H_{uu}^{-1} (H_{ux} + f_u^T v_{xx}) \\
 \beta_2(t) &= -H_{uu}^{-1} f_u^T v_{xb}
 \end{aligned}$$

$$s_3(t) = -H_{uu}^{-1} f_u^T v_{xt_f} \quad (66)$$

In Eq (64) the total number of differential equations to be integrated exceeded 130 for this problem. The V_{xx} equations require the second partial derivatives of the Hamiltonian with respect to the states, and several of these equations were in excess of 15 lines of fortran computer coding.

In addition to the integration of the above equations, the terminal conditions had to be computed for each equation. These computations required the use of a numerical differentiation technique as was outlined in Chapter VI. It should be noted that this procedure requires a significant amount of calculation in order to obtain the numerical derivatives for these terminal condition values. Along with these calculations, an iteration was required to solve the transcendental equation for the optimizing control, as discussed in Chapter VI.

When attempting to run this program on the computer the size of the program was such that the core memory allocation requirement was in excess of 177,000 (octal) words. The corresponding compilation time for each run was one hundred thirty-two seconds.

In order to reduce the execution time the initial flight interval was limited to one second with the intention of increasing the limit when the program was working properly. The results to this point indicate that the state equations have been partially successful in both the backward and

forward integration for the one second time interval. However, difficulty was encountered during the backward integration of the costate equations which determine the sensitivity coefficients. The backward and forward integration of the state equations plus the initial backward integration of the costate equations required over fifty seconds execution time.

Due to the large memory requirement and the lengthy execution time, this program was placed near the bottom of priorities in the central computer. Turn-around time for each computer run varied between 2 and 3 days.

In addition to the slow computer turn-around time, there exists within the program itself the inherent problem associated with debugging the complex program code. That is, there exists some probability of programming errors even though every attempt was made to limit such a possibility. The complexity of the equations used in this program definitely contributes to the possibility of a mistake in derivation of the equations.

This investigation demonstrates the desirability of finding an alternate method of generating the equations required for the second order optimization technique. The alternate method should not require the lengthy derivation of partial derivative equations nor the large amount of memory required to compute them.

A procedure that satisfies these requirements is a higher order computer language designated PROSE, Ref (2:1). It evaluates automatically partial derivatives as a by-product of

the computations in the model. The resulting derivative values are exact to the precision of the computer, as though they had been evaluated by formula derivation and evaluation. This derivative evaluation by PROSE would eliminate the lengthy, time consuming, and possibly erroneous derivation of the partial derivatives.

VIII. Conclusions and Recommendations

The method of solution was one of the primary concerns in this study. The rate of convergence using a first order optimization technique for solutions to this missile evasion problem was estimated to be extremely slow and, therefore, unacceptable. The second order technique used for this program also has disadvantages; the two primary ones are long computer execution times and the large number of equations that had to be derived. The following procedures are recommended as additional efforts in solution methodology:

1. Simplify the problem formulation as follows:
 - a. Use an altitude difference between the missile and target as one state instead of the state for each missile altitude and target altitude.
 - b. Assume the time responses for the angle of attack of the target and the coefficient of thrust of the target are rapid enough to neglect any time delay in the response.
- With the above simplifications incorporated into the program the state vector would be reduced to nine states.
2. Complete the second order optimization routine as derived in this thesis in order to obtain the optimum control strategies.

3. Investigate use of the PROSE programming language for solving the evasion problem and obtaining the optimum control strategies.
4. Compare the results obtained with the optimization program derived for this thesis to those obtained with PROSE.

Bibliography

1. Jacobson, D. H., and D. Q. Mayne. Differential Dynamic Programming. New York: American Elsevier Publishing Company, Inc., 1970.
2. Control Data Corporation. PROSE A General Purpose Higher Level Language. Calculus Applications Guide. Minneapolis Minnesota: Control Data Corporation, Jan. 1977.

Appendix A

Detailed Schematic of the Program

A detailed flow chart of steps 1 through 6 used in the program is illustrated in Fig. 10. The initial values of the variables MK, C, b, and E are annotated at the top of the diagram. Recall that the initial value for u_{nom} is given as the following:

$$\begin{aligned} u_1 &= 0. \\ u_2 &= 0. \\ u_3 &= .025 \end{aligned} \tag{96}$$

where the controls are assumed constant throughout the entire time of flight.

NEFF(2) is the point at which A exceeds the present value of .0005; and in the calculation of db and dt_f , recall that these values are determined by the following:

$$\begin{bmatrix} db \\ dt_f \end{bmatrix} = -E \begin{bmatrix} v_{bb} & v_{bt_f} \\ v_{t_fb} & v_{t_ft_f} \end{bmatrix}^{-1} \begin{bmatrix} v_b \\ v_{t_f} \end{bmatrix} \tag{90}$$

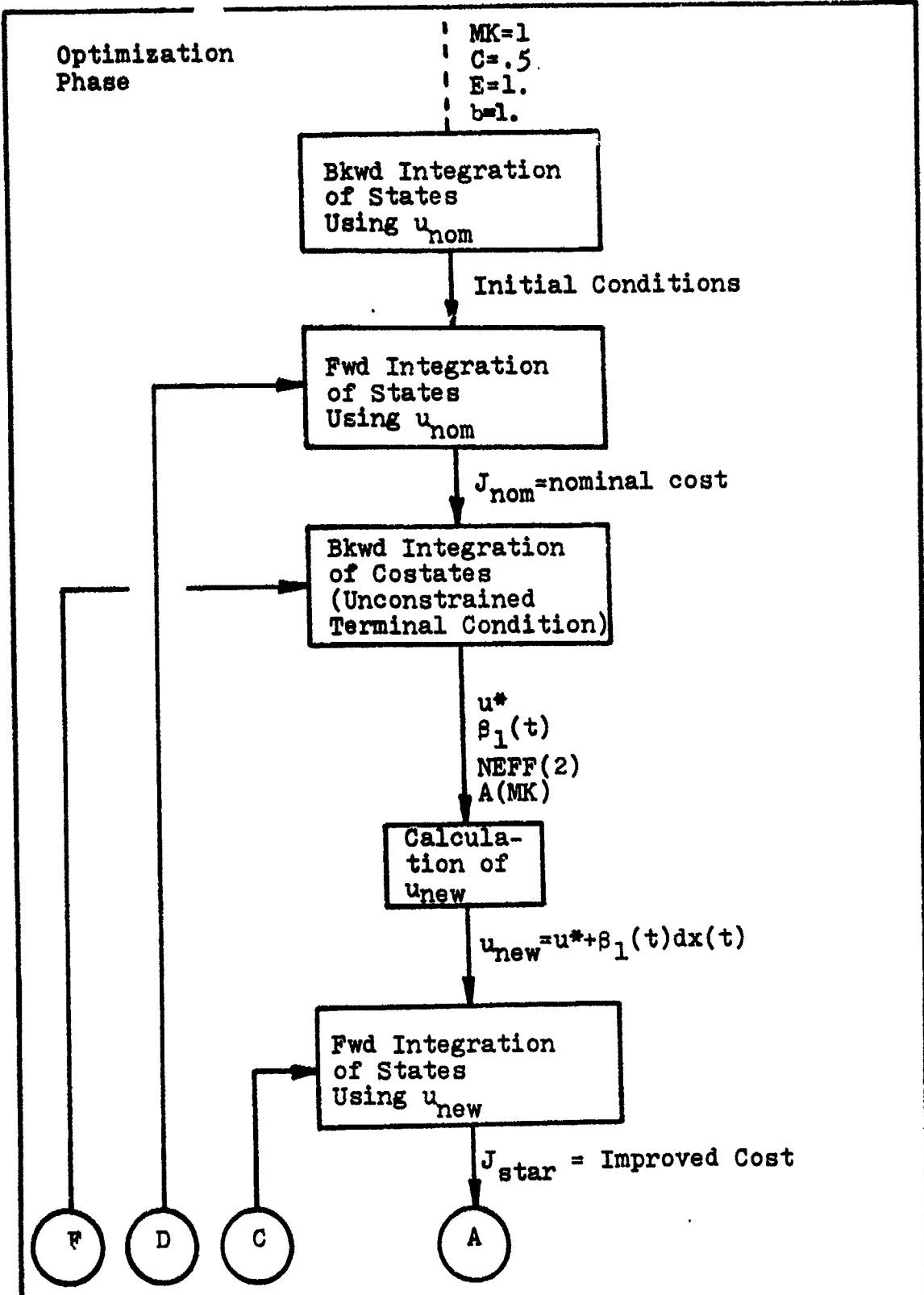


Fig. 10. Detailed Flow Chart of the Program

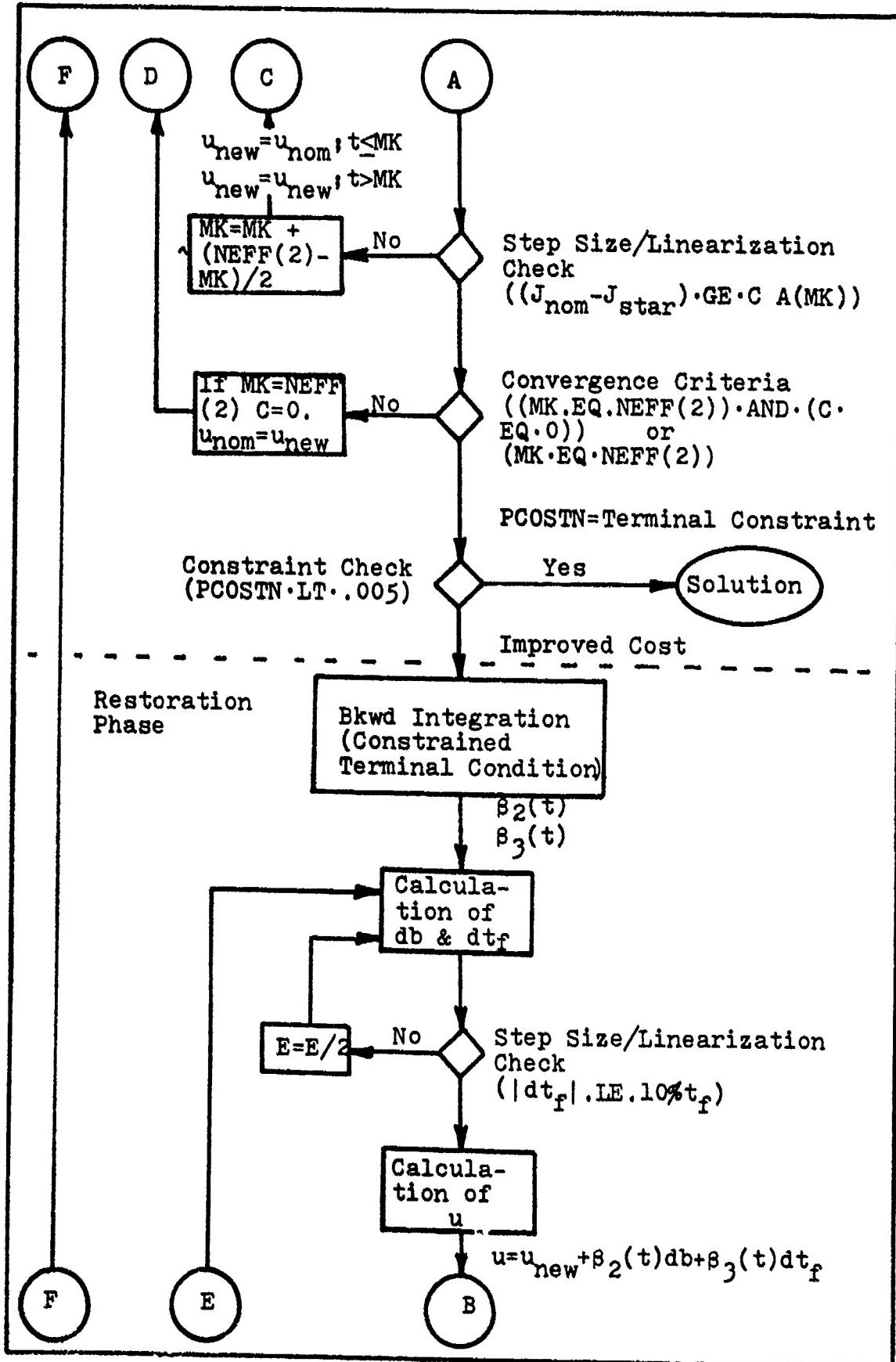


Fig. 10. Detailed Flow Chart of the Program
(Continued)

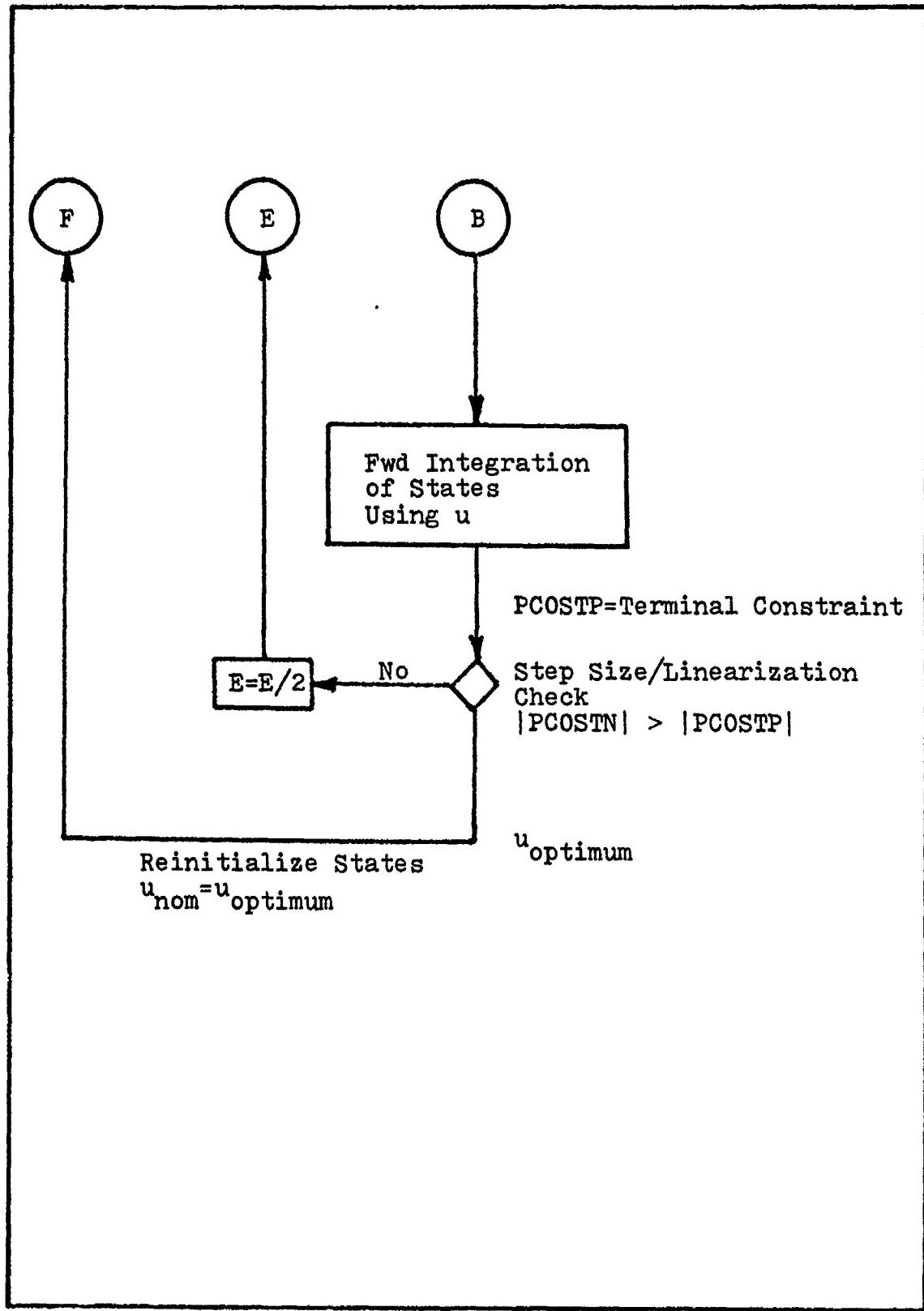


Fig. 10. Detailed Flow Chart of the Program
(Concluded)

VITA

Robert Smith was born on 11 September 1943 in Seattle, Washington. He graduated from high school in Renton, Washington in 1961 and attended Everett Junior College from which he received the Associate of Arts and Science degree in June 1963. He then attended the University of Washington where he received the Bachelor of Science degree in Aeronautical and Astronautical Engineering in June 1966. Upon graduation he was employed as an aerospace engineer for the Boeing Company, Seattle, Washington until selected for Officers Training School in August 1968. He completed pilot training and received his wings in December 1969. He then served as an F-106 pilot at Griffiss AFB, New York until going on an overseas tour to Iceland in the F-102. Upon his return he again served as an F-106 pilot at Castle AFB, California until entering the School of Engineering, Air Force Institute of Technology, in June 1976.

Permanent address: 1008 N 33rd PL

Renton, Washington

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GA/EE/77-5	2. GOVT ACCESSION NO.	3. DESCRIPTIVE CATALOGUE NUMBER <i>Master's Thesis</i>
4. TITLE (and Subtitle) OPTIMAL MISSILE EVASION		5. TYPE OF REPORT OR TECHNICAL DATA SOURCE MS Thesis
6. AUTHOR(s) Robert L. Smith	7. PERFORMING ORG. REPORT NUMBER	
8. CONTRACT OR GRANT NUMBER(s)		9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ASDP 08342260N		11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Development Planning (ASD/XRHA) Aeronautical Systems Division Wright-Patterson AFB, Ohio 45433
12. NUMBER OF PAGES 16 p.		13. SECURITY CLASS. (of this report) Unclassified
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		
15. DECLASSIFICATION/DOWNGRADING SCHEDULE		
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Approved for public release; IAW AFR 190-17 JEREMY F. GUSS, DAPT, USAF Director of Information		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Missile Evasion Optimum Control Aircraft Survivability Missile Launch Envelopes		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The purpose of the study is to formulate a method to determine the control strategies that maximize the probability of survival for an evading aircraft against an air-to-air missile. The missile model developed is a typical infrared missile using proportional navigation steering. The P_k is modeled as ellipsoidal iso-cost surfaces with a cost value that decays exponentially. The problem terminates when the line-		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Block 20 - Continued

off-sight from the missile to the target aligns on the edge of the missile's fuzing cone angle. The algorithm developed employs a second order differential dynamic programming model which optimizes the controls of the evading aircraft.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)